

Treball de Fi de Màster

Màster Universitari Enginyeria Industrial

**Resolució del Parallel Assembly Line Balancing Problem
(PALBP) considerant dues línies de muntatge**

MEMÒRIA

Autor: Axel Felis Llena
Director: Alberto García Villoria
Convocatòria: Octubre 2016



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Els fabricants utilitzen les línies de muntatge per produir un gran volum d'un producte. Una línia de muntatge és una seqüència d'estacions de treball connectades entre si a través d'un sistema de manipulació de material. Aquest sistema, s'utilitza per unir diferents subproductes per arribar així a un producte final. El problema de l'equilibrat d'una línia de muntatge és un problema clàssic en l'enginyeria industrial. La investigació va començar a mitjans dels anys 1950 i dècada dels 60 i encara ara es continua investigant sobre l'equilibrat de les línies de muntatge, l'estructura bàsica del problema és rellevant per al disseny de sistemes de producció d'avui en dia, tant en plantes automatitzades com manuals.

El problema de l'equilibrat de la línia de muntatge és l'assignació de tasques a les estacions de treball per tal d'optimitzar el temps d'inactivitat de la línia o del treballador. La primera condició és que el temps de treball total assignat a cada estació de treball ha de ser inferior a o igual que el temps de cicle (interval de temps per realitzar un conjunt de tasques en una estació). La segona, és que l'assignació de tasques ha de seguir l'ordre depenent de les successores i predecessores que tingui cadascuna d'elles.

L'objectiu d'aquest projecte és la resolució, mitjançant mètodes heurístics (no exactes), del problema d'equilibrat de línies de muntatge en paral·lel o Parallel Assembly Line Balancing Problem (PALBP).

La resolució del PALBP es farà considerant el cas de dues línies de muntatge en paral·lel amb un producte assignat a cadascuna i considerant que el temps de cicle per un producte és el mateix o el doble que el de l'altre producte. Els operaris poden ser assignats a una estació de treball d'una línia, o compartit entre dues estacions veïnes de les dues línies. Tanmateix, es poden deixar estacions buides (sense cap operari). Tenint en compte que el nombre màxim d'estacions per línia és limitat, l'objectiu del treball serà minimitzar el nombre d'operaris total entre les dues línies de muntatge respectant els temps de cicle i les relacions de precedències entre tasques. Per tal de poder resoldre el problema s'estudiarà la literatura científica i es proposarà diferents procediments de resolució del PALBP. Els procediments proposats s'implementaran i es farà una extensa experiència computacional per validar-los.

Concretament es proposaran dos procediments diferents per resoldre el problema d'equilibrat de línies de muntatge en paral·lel. Els dos procediments que es proposen intentaran trobar la millor solució possible. Per fer-ho, s'utilitzaran alguns elements comuns però el mètode serà totalment diferent.

Sumari

RESUM	1
SUMARI	3
1. GLOSSARI	5
2. INTRODUCCIÓ	6
2.1. Objectius del projecte	7
2.2. Abast del projecte	7
3. INTRODUCCIÓ A LES LÍNIES DE MUNTATGE	8
3.1. Les línies de muntatge	8
3.1.1. Introducció històrica	8
3.1.2. Conceptes bàsics	8
3.1.3. Classificació de les línies de muntatge	9
3.2. Els problemes de línies de muntatge	15
3.2.1. Disseny de les línies (ALDP)	16
3.2.2. Equilibrat de les línies (ALBP)	16
4. PARALLEL ASSEMBLY LINE BALANCING PROBLEM (PALBP)	20
4.1. Característiques principals del problema a resoldre	20
4.2. Algoritmes per al PALBP	22
4.2.1. Mètodes exactes	22
4.2.2. Mètodes no exactes	23
5. RESOLUCIÓ DEL PALBP	29
5.1. Elements comuns	29
5.1.1. Fita mínima	29
5.2. Heurística Ad hoc	29
5.2.1. Funcionament de l'heurística Ad hoc	29
5.2.2. Heurística Ad hoc	30
5.2.3. Reparació de la solució	34
5.2.4. Exemple de resolució de la heurística Ad hoc	37
5.3. Simulated Annealing (SA)	43
5.3.1. Càlcul de la solució inicial	43
5.3.2. Millora de la solució inicial	44
5.4. Preprocés	46
6. EXPERIÈNCIA COMPUTACIONAL	48

6.1. Els exemplars de testeig.....	48
6.2. Anàlisi dels resultats	48
7. PRESSUPOST	52
8. IMPACTE AMBIENTAL	55
CONCLUSIONS	56
AGRAÏMENTS	57
BIBLIOGRAFIA.....	58
Referències bibliogràfiques	58
ANNEX 1: RESULTATS	60
ANNEX 2: DOCUMENTACIÓ ADJUNTA.....	63

1. Glossari

A continuació es mostra un llistat amb les diferents abreviacions que apareixen al llarg del treball.

n_j	total de tasques del producte j
P	Conjunt de productes; $P=(A \text{ i } B)$
J_p	conjunt de tasques per cada producte. Així, una tasca està definida per la parella (p, i) del producte $p \in P$ i les seves tasques $i \in \{1, \dots, J_p\}$
K	nombre màxim de columnes d'estacions per línia. Una columna d'estacions està definida per una parella d'estacions.
h_p	línia assignada al producte $p \in P$
T_k	conjunt de tasques (p, j) assignades al operari col·locat en una estació (h_p, k)
$NIS_{i,j}$	nombre de successors immediats de la tasca i del producte j
$NS_{i,j}$	nombre de successors totals de la tasca i del producte j
$t_{i,j}$	temps de procés de la tasca i del producte j
$S_{i,j}$	conjunt de tasques successors totals de la tasca i del producte j
$P_{i,j}$	conjunt de tasques predecessors totals de la tasca i del producte j
$E_{i,j}$	earliest estació en que pot començar la tasca i del producte j
$L_{i,j}$	lastest estació en la que pot començar la tasca i del producte j
TC_j	temps de cicle del producte j
C_{\min}	fita mínima
α	paràmetre a calibrar d'un dels indicadors utilitzats
β	paràmetre a calibrar d'un dels indicadors utilitzats
γ	paràmetre a calibrar d'un dels indicadors utilitzats
σ	paràmetre a calibrar d'un dels indicadors utilitzats
μ	paràmetre a calibrar d'un dels indicadors utilitzats
θ	paràmetre a calibrar d'un dels indicadors utilitzats
$UT(\Delta)$	conjunt de tasques sense assignar

2. Introducció

En una línia de muntatge (o de producció, indistintament) un conjunt de peces han de ser muntades per produir un producte intermedi o final. Per realitzar aquest procés es disposa d'un nombre limitat d'estacions, en les quals s'ha de desenvolupar un conjunt determinat de tasques. Cada tasca té una durada determinada, les tasques tenen relacions de precedència entre elles i suposen unitats indivisibles, és a dir, una tasca no es pot dur a terme en dues estacions diferents. Les precedències entre tasques es poden representar mitjançant un graf (figura 1), on els vèrtexs simbolitzen les tasques i els arcs les relacions de precedència entre una tasca i una altra.

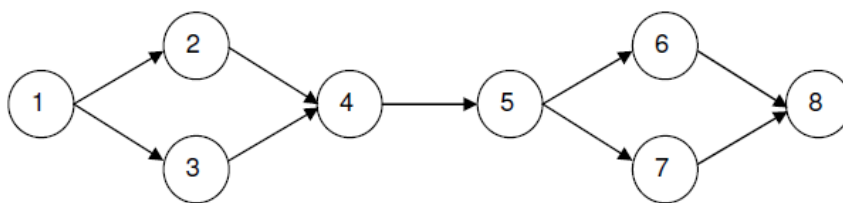


Figura 1: Exemple de graf. [FONT: pròpia]

El problema PALBP (Parallel Assembly Line Balancing Problem) consisteix en equilibrar un nombre de línies de muntatge disposades en sèrie de forma paral·lela. Els diferents productes es poden assignar a diferents línies, considerant un temps de cicle per a cada una de les estacions (per temps de cicle entenem el temps màxim que pot estar l'operari realitzant tasques en una estació). Tot i que el concepte de línies de muntatge paral·leles es conegut tant a nivell industrial com acadèmic, no hi ha gaires treballs de recerca sobre el seu equilibrat en la literatura existent.

La majoria de les plantes de fabricació consisteixen en una o més línies de muntatge. Quan la demanda és alta, una opció possible és duplicar la cadena de muntatge, i així, poder produir més. Un altre dels avantatges de tenir línies paral·leles de muntatge es veu durant les avaries d'estacions de treball. Si es produeixen problemes en els equips d'una estació de treball, altres línies poden seguir funcionant, en canvi, una única línia ha de ser apagada quan es produeix una avaria en qualsevol estació de treball.

Millorar la productivitat dins les línies de muntatge és molt important perquè permet augmentar la capacitat i reduir costos. La literatura està plena d'investigacions sobre l'equilibratge de línies individuals, però en la realitat existeixen multitud d'empreses que disposen de línies paral·leles que permeten la compartició d'operaris.

2.1. Objectius del projecte

L'objectiu d'aquest problema és millorar la resolució del PALBP mitjançant procediments heurístics (no exactes). En aquest problema es desitja reduir el màxim possible el nombre d'operaris que hi ha en dues línies de muntatge en paral·lel i per tant augmentar al màxim l'eficiència de la línia.

Per fer-ho, es presentaran un conjunt de procediments que s'han resolt per altres autors per obtenir fites i solucions factibles inicials del problema i és solucionarà el problema amb dos procediments amb els que fins ara no s'havia resolt aquest problema.

Per tal d'avaluar l'eficiència del procediment de resolució proposat, es presentarà una experiència computacional realitzada a partir d'uns exemplars de testatge, de la qual se n'analitzaran els resultats obtinguts.

2.2. Abast del projecte

En aquest projecte es tracta la resolució del PALBP des d'una perspectiva teòrica i d'investigació. Concretament, la variant estudiada té en compte les següents hipòtesis i característiques:

- El nombre línies és conegut, amb un total de 2 línies on en cada una d'elles hi ha un producte diferent.
- El temps de cada una de les tasques és conegut.
- Es coneixen totes les tasques de cada un dels productes.
- Hi ha un temps màxim d'execució de tasques per cada estació (temps de cicle), que és el mateix en totes les columnes d'estacions.
- Les tasques són indivisibles i no es poden interrompre un cop iniciada la seva execució.
- Existeixen relacions de precedència entre tasques.
- La línia de muntatge és serial però amb dues línies en paral·lel.
- La taxa d'entrada i de sortida de les unitats a la línia és fixa i coincideix amb el temps de cicle.
- Es coneix el nombre d'estacions de treball màxim.
- Els operaris poden realitzar tasques de les dues línies en una mateixa columna d'estacions.

3. Introducció a les línies de muntatge

3.1. Les línies de muntatge

3.1.1. Introducció històrica

Les línies de muntatge van ser introduïdes en l'era industrial (s. XVIII) per Eli Whitney, el qual va inventar el sistema de manufactura americana el 1799, utilitzant el concepte de la divisió del treball i tolerància en enginyeria, per tal de crear assemblatges de parts d'una manera repetitiva. No obstant això, la idea de línies de muntatge no era nova, ja que aquesta va ser desenvolupada a Venècia centenars d'anys abans, on els vaixells eren produïts en massa utilitzant parts que ja havien estat manufacturades prèviament. L'arsenal de Venècia produïa quasi un vaixell per dia, estem parlant doncs del que podria considerar-se una de les primeres línies més importants del món.

Posteriorment, els coneixements assolits en les línies de muntatge juntament amb els avenços tecnològics van assentar les bases de la producció en cadena. L'any 1913, Henry Ford va instal·lar la primera línia de muntatge mòbil amb la idea de disminuir costos i permetre d'aquesta manera la producció massiva.

3.1.2. Conceptes bàsics

Aquest apartat està basat principalment en l'article de Pinto et al. (1983), el capítol I de Scholl (1999) i l'article de Becker i Scholl (2006):

A continuació es presenten els conceptes més rellevants dels problemes de línies de muntatge:

- **Tasca i :** és una unitat de treball indivisible que té associat un temps de procés (t_i). El treball total necessari per fabricar una unitat de producte en una línia es divideix en un conjunt de N tasques.
- **Temps de procés de la tasca i (t_i):** és el temps necessari per fer la tasca i .
- **Relacions de precedència:** estan definides per les restriccions sobre l'ordre en el qual es poden realitzar les tasques en la línia de muntatge. Així, una tasca no pot ser realitzada fins que no s'hagin realitzat totes les seves tasques precedents. Aquest conjunt de les relacions de precedència es pot representar en forma d'un graf de precedències.

- **Estació de treball k :** és la part de la línia de muntatge en la que es realitzen les tasques. Les estacions de treball poden ser de diferents tipologies: des de qualsevol tipus de maquinària fins a mecanismes especialitzats. Els operaris que s'hi assignen poden ser humans o robots, depenent del tipus de línia que s'estigui tractant. En una línia de muntatge hi ha K estacions de treball ($k = 1, \dots, K$).
- **Càrrega de treball de l'estació k :** és la suma dels temps de procés de totes les tasques assignades a l'estació k . En el cas que es considerés un temps de preparació entre tasques, aquest es tindria en compte en el càlcul de la càrrega de treball (Boysen et al., 2007). En alguns problemes d'equilibrat de línies, també se suma en funció de la seqüència en què es realitzin les tasques, un cert temps de procés associat al temps de preparació de les eines, moviments de l'operari, rotació de les peces, etc.
- **Temps de cicle (TC):** és el temps disponible a cada estació per completar les tasques assignades per a cada unitat de producte. En cas que no estigui especificat, s'acostuma a considerar igual a la major de les càrregues de treball d'entre totes les estacions.
- **Temps mort o temps ociós de l'estació k :** és la diferència entre el temps de cicle i la càrrega de l'estació k .
- **Eficiència de la línia (η):** és el sumatori dels temps de procés de totes les tasques dividit pel temps total disponible, essent aquest el producte del temps de cicle pel nombre d'estacions que hi ha a la línia ($\eta = \frac{\sum_{i=1}^N t_i}{TC \cdot K}$).

3.1.3. Classificació de les línies de muntatge

En la literatura les línies de muntatge es classifiquen de diverses formes, depenent de diferents conceptes com ara l'arquitectura de la línia, tipus de producte, durada de les tasques, tipus d'operador, disciplina d'entrada de peces i flux de peces. A continuació es resumeixen les principals classificacions proposades per Ghosh i Gagnon (1989), Plans (1999), Scholl (1999) i Becker i Scholl (2006).

A. Tipus de línies de muntatge segons la seva arquitectura:

- ❖ **Línia serial:** es tenen estacions simples col·locades en sèrie, per on les unitats passen consecutivament d'una estació a la següent. En aquest tipus de línies el transport no el solen fer operaris sinó diferents mecanismes com podria ser una línia transportadora.

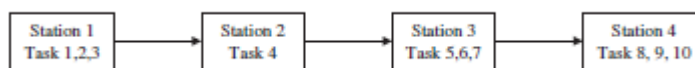


Figura 2: Exemple línia de muntatge tipus serial. [FONT: pròpia]

- ❖ Línia amb estacions en paral·lel: consisteix en una única línia on les estacions en paral·lel estan permeses. Les estacions en paral·lel consisteixen en dues o més estacions idèntiques que realitzen en paral·lel les mateixes tasques. Amb aquest tipus d'arquitectura s'aconsegueix solucionar el problema que es crea amb aquelles tasques amb un temps de procés major al temps de cicle, ja que d'aquesta manera, la durada d'una tasca es redueix proporcionalment al nombre d'estacions en paral·lel. Per exemple, suposant que una tasca té una durada de 30 segons però el temps de cicle es de 20 segons, en el punt on és realitzés aquella tasca es col·locaria una estació paral·lela per permetre acabar la tasca reduint el temps mig a 15 segons.



Figura 3: Exemple línia de muntatge amb estacions en paral·lel. [FONT: propia]

- ❖ Línies paral·leles: són diverses línies col·locades en paral·lel. El que permet és realitzar diferents productes en paral·lel i fer que el nombre d'operaris sigui inferior a si es tractessin els productes per separat. A més a més una característica molt important és que els operaris poden treballar en dues estacions sempre i quan siguin de la mateixa columna i contigües.

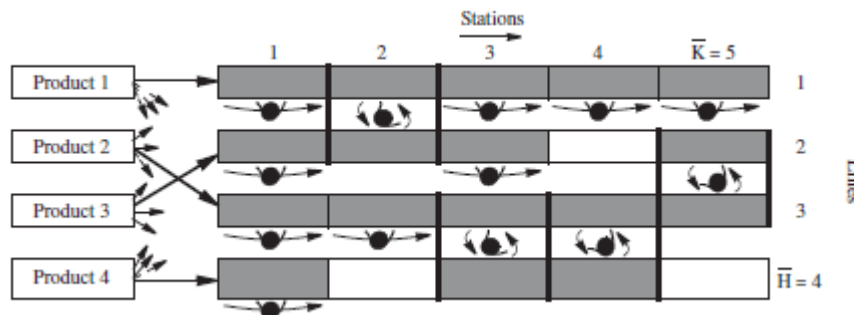


Figura 4: Exemple línia de muntatge tipus paral·lel. [FONT:[6]]

- ❖ Línies de dos costats: són dues línies serials en paral·lel, on en comptes de tenir una estació simple, es tenen dues estacions oposades a cada costat de línia (estació dreta i esquerra) que processen simultàniament una peça comuna. Es poden utilitzar en qualsevol tipus de producte, però són més comuns en els productes grans, on es requereix desenvolupar tasques en ambdós costats, com és el cas dels cotxes.

D'aquesta manera, algunes tasques han de ser assignades a un dels costats de la línia (per exemple, posar la tapa del dipòsit de benzina), altres poden ser assignades de manera indiferent a qualsevol dels dos costats (posar el maleter) i altres han de ser assignades a banda i banda simultàniament (posar les rodes). D'aquesta manera, el temps total es inferior, ja que l'operari només s'ha de centrar en uns dels dos costats i no ha de perdre temps amb desplaçaments.

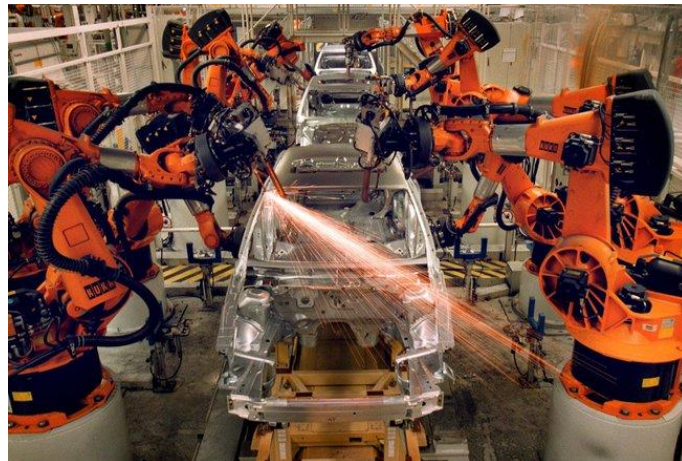


Figura 5: Exemple muntatge cotxe en una línia de dos costats. [FONT: www.noticias.coches.com]

- ❖ Línies circulars o tancades: en aquest cas es té una cinta circular o tancada que serveix a totes les estacions. En aquesta, les peces circulen mentre són preses pels operaris, i un cop han acabat de processar les peces, les alliberen de nou a la cinta, excepte l'estació on es realitza l'última tasca, que diposita la peça fora de la cinta.

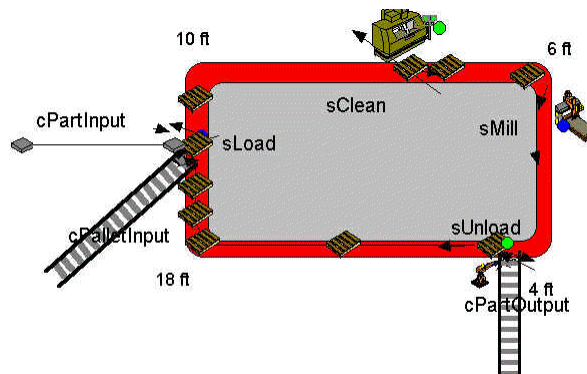


Figura 6: Exemple línia de muntatge de tipus circular. [FONT: wisc.edu]

- ❖ Línies en forma d'U: les línies en forma d'U són conseqüència de la introducció del concepte de producció *just in time*. Aquesta disposició de les estacions fa més flexible el sistema de producció, ja que les estacions poden treballar en dos segments de la línia de muntatge si tots dos estan un davant l'altre. Aquesta configuració pot donar un

millor balanç de les càrregues de les estacions, ja que el nombre de possibles assignacions tasques-estació és més gran. A més a més,

es pot donar el cas de que hi hagi estacions en forma d'U en paral·lel.



Figura 7: Exemple línia de muntatge en forma d'U. [FONT: slideshare]

B. Tipus de línies de muntatge segons el seu flux de les peces:

- ❖ Línies síncrones: totes les estacions tenen un temps de cicle comú. Conseqüentment, les peces passen d'una estació a la següent al mateix temps, de manera que no és necessària la instal·lació de pulmons o *buffers* entre estacions per esperar que la tasca posterior s'hagi realitzat.
- ❖ Línies asíncrones: les estacions estan relacionades a través de pulmons o *buffers* que es col·loquen entre elles per emmagatzemar les peces que s'han processat en les estacions precedents. En aquest cas, les estacions tenen velocitats de procés diferents. Aquestes línies comporten un problema: prendre la decisió sobre on col·locar els *buffers* i la mida que aquests han de tenir. En resum, els *buffers* serveixen d'emmagatzematge de productes semi acabats que se situen al mig de les línies de muntatge.
- ❖ Línies d'alimentació: són sistemes de muntatge composts en què es té una o més línies d'alimentació o *feeder lines*, on es realitzen sub-acoblats, és a dir, es té una línia principal on es van muntant les peces, i línies suplementàries que pre-processen certes parts per, posteriorment, alimentar amb productes intermedis la línia de muntatge principal. El problema afegit que comporta aquest tipus de línia és que cal sincronitzar la taxa de producció de les línies d'alimentació. Un exemple clar, seria el muntatge d'un avió on a mesura que es va muntant, apareixen diferents parts com el motor el qual ha estat muntat anteriorment.

C. Tipus de línies de muntatge segons el tipus de productes que treballen:

- ❖ Línies simples: processen un únic tipus de producte, i les estacions realitzen repetidament les mateixes tasques.
- ❖ Línies mixtes: són línies que permeten produir simultàniament diferents models. Les tasques a realitzar són molt semblants per cadascun dels models. La producció en aquestes línies no necessita temps de preparació entre models (o és molt reduït) a causa de la semblança de les operacions.
- ❖ Línies multi-model: en aquesta tipologia varien els processos de producció d'un model a un altre. Per consegüent, es produeix per lots i es tenen en compte els temps de preparació de cada estació de la línia per canviar d'un producte a un altre.

D. Tipus de línies de muntatge segons la duració de les tasques:

- ❖ Línies deterministes: el temps de procés de les tasques es coneix amb certesa i no pot variar.
- ❖ Línies estocàstiques: el temps de procés de les tasques és aleatori o probabilístic, perquè la variabilitat intrínseca del mateix és significativa. Així, es considera que el temps de procés segueix una determinada llei estadística.
- ❖ Línies dependents: el temps de procés de les tasques depèn d'alguna variable, com per exemple l'estació en la qual es processa, la seqüència que se segueix, l'operador que la realitza, l'instant en què es realitza, etc.

Ghosh i Gagnon (1989, p.637-670), varen presentar de forma esquemàtica una classificació de les línies de muntatge segons els conceptes definits als apartats C i D (Figura 8).

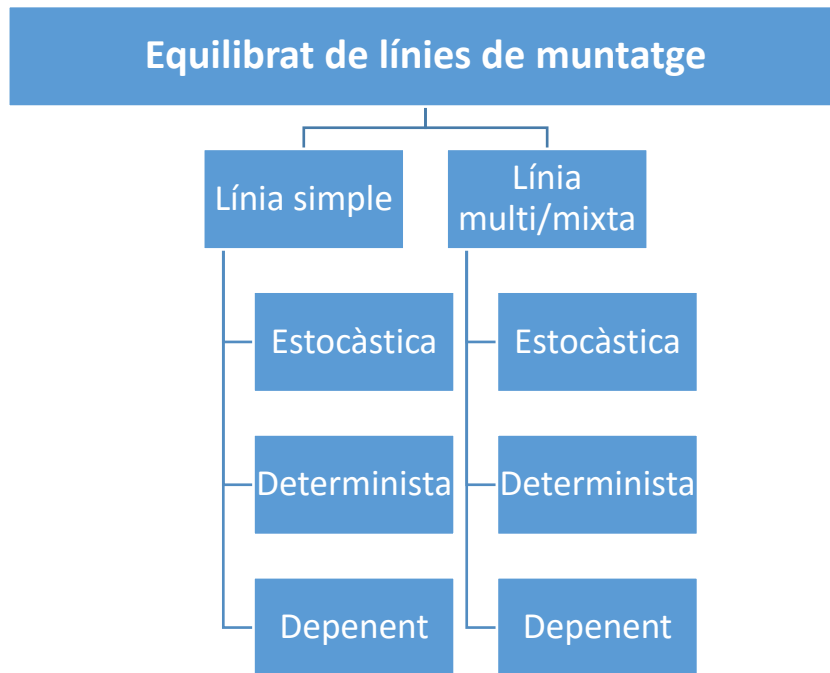


Figura 8: Classificació de Glosch y Gagnon. [FONT: Glosch i Gagnon 1989]

E. Tipus de línies de muntatge segons la disciplina d'entrada de les peces a la línia:

- ❖ Línies d'entrada fixa: les peces arriben a intervals regulars. En el cas particular en què la línia és síncrona, aquest interval es correspon amb el temps de cicle.
- ❖ Línies d'entrada variable: les peces arriben a intervals variables. Normalment les arribades segueixen una determinada llei estadística.

F. Tipus de línies de muntatge segons el tipus d'operari de les estacions:

- ❖ Línies manuals: els operadors són humans, independentment que els processos siguin automatitzats o no.
- ❖ Línies robotitzades: els operadors són robots i, conseqüentment, els processos estan totalment automatitzats.

A continuació es mostra un resum dels diferents tipus de línies explicats anteriorment (figura 9):

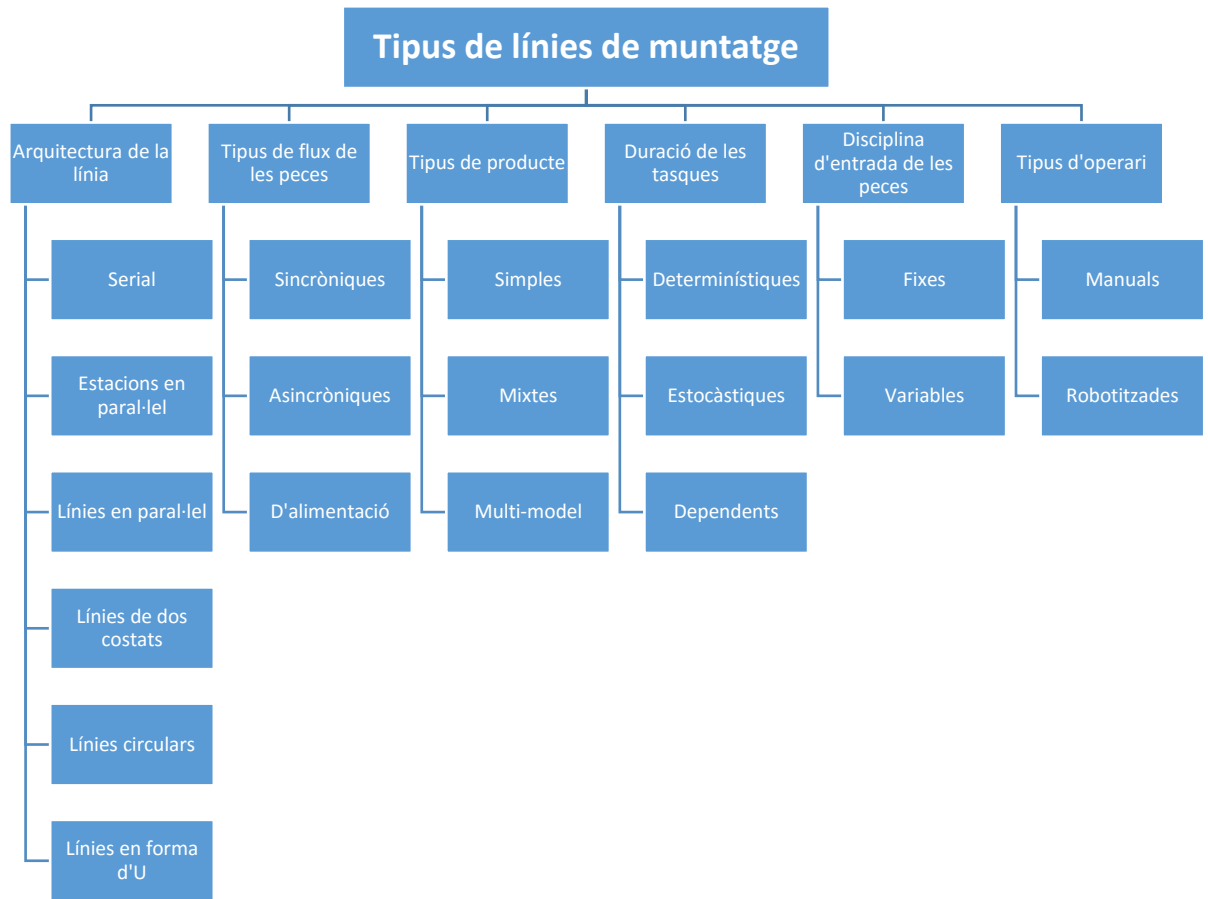


Figura 9: Resum tipus de línies de muntatge. [FONT: pròpia]

3.2. Els problemes de línies de muntatge

Els problemes de línies de muntatge es classifiquen en funció de l'objectiu que es vol assolir i les restriccions que hi són presents.

Els problemes més comuns presents a la literatura són els problemes de disseny de línies de muntatge (ALDP, Assembly Line Design Problem) i els problemes d'equilibrat de línies de muntatge (ALBP, Assembly Line Balancing Problem).

No obstant això, també es poden trobar problemes menys habituals que consisteixen a equilibrar línies de desmuntatge (DLBP, Disassembly Line Balancing Problem). En el cas dels DLBP, tal com indica el seu nom, es té un cert producte muntat i es vol decidir l'ordre d'execució de les tasques que permeti desmuntar-lo eficientment.

A continuació s'expliquen els dos tipus de problemes de línies més comuns: el ALDP i el ALBP.

3.2.1. Disseny de les línies (ALDP)

El disseny d'un sistema productiu és, en general, un procés complex que requereix un gran nombre de decisions interrelacionades amb repercussions sobre objectius múltiples. D'això se'n desprenen, almenys, dues conseqüències importants.

En primer lloc, que el procés de disseny, excepte en casos molt particulars, és difícil de formalitzar i d'automatitzar. Requereix la intervenció de persones, que prenen decisions basant-se en els seus coneixements i la seva experiència i, en alguns casos, amb la informació que els proporcionen algorismes d'optimització o de simulació. Així doncs, en un procés de disseny una o diverses persones, seguint els passos establerts en un mètode formalitzat amb graus molt variables de precisió, interactuen amb les eines de què disposen.

En segon lloc que, tot i que per aconseguir una solució òptima totes les decisions de disseny i equilibrat s'han d'adoptar de forma simultània i no seqüencial, en la pràctica s'aplica un procediment jeràrquic, en què unes decisions es prenen abans que altres i les condicionen.

A l'hora de dissenyar la línia es tenen en compte conceptes com les dimensions dels espais d'emmagatzematge i els pulmons o *buffers*; els sistemes de transport utilitzats a la línia; els equips de manufactura utilitzats per a cada tasca; etc.

Diferents factors han generat nous problemes de disseny. Aquests factors són, entre altres, l'actualització dels sistemes de producció caracteritzats per una vida útil curta, un alt grau d'automatització, l'aparició de nous equips de treball per dur a terme la manufactura, noves tecnologies, etc. (Rekiek et al., 2002).

3.2.2. Equilibrat de les línies (ALBP)

L'objectiu d'aquest tipus de problema és assignar les tasques que són necessàries per al muntatge d'un producte a les diferents estacions que es tenen disponibles, en la corresponent línia de muntatge. A més a més aquestes tasques tenen diferents relacions de precedència que permeten crear diferents solucions.

Segons el plantejament clàssic de l'ALBP, l'objectiu és trobar el nombre mínim d'estacions, o el temps de cicle mínim, necessari per processar les tasques, complint les relacions de precedència.

Segons Baybars (1986, p.909-932), una línia està ben equilibrada si en plena utilització dels seus recursos, els temps lliures o ociosos de cada estació sumats són mínims. Baybars

estableix a més que si s'aconsegueix que totes les estacions tinguin el mateix temps d'execució, l'equilibri és perfecte tot i que aquest objectiu és diferent d'aconseguir el nombre d'operaris mínims.

A la literatura es poden trobar diverses classificacions de problemes d'ALB, les més conegudes i acceptades històricament, són les següents:

- Baybars (1986) distingeix dos tipus de problemes: el simple (SALBP: Simply Assembly Line Balancing Problem) i el general (GALBP: General Assembly Line Balancing Problem). El SALBP acostuma a ser una versió simplificada i acadèmica (tot i així molt difícil de resoldre òptimament) dels problemes reals, mentre que el GALBP té en compte característiques més complexes que es troben en les línies reals.
- Ghosh i Gagnon (1989) distingeixen quatre tipus de problemes: el simple determinista, el simple estocàstic, el multi/mixt determinista i el multi/mixt estocàstic.

A continuació s'expliquen les característiques dels problemes més importants.

A. SALBP: Simply Assembly Line Balancing Problem

Aquest tipus de problema abasta els problemes d'equilibrat més simples, que les seves característiques són:

- ✓ Producció en massa d'un únic producte homogeni.
- ✓ Cada tasca és indivisible.
- ✓ La línia és simple.
- ✓ La línia és serial.
- ✓ Les tasques tenen entre elles restriccions de precedència.
- ✓ Les dades són conegudes i deterministes, normalment es disposa d'un nombre d'estacions o temps de cicle fixat.
- ✓ La taxa d'entrada de peces a la línia és fixa.
- ✓ Totes les estacions estan equipades de forma equànime tant en màquines com en treballadors.

A partir d'aquestes característiques comunes, es poden distingir quatre variants del SALBP:

- a) SALBP-1: es disposa d'un temps de cicle màxim assignat (o taxa de producció) i es busca minimitzar el nombre d'estacions necessàries per realitzar el procés. Aquest tipus de problema és comú quan la demanda externa pot ser estimada i es vol instal·lar un nou sistema de muntatge.
- b) SALBP-2: en aquest cas, es parteix d'un nombre d'estacions fix i es busca minimitzar el temps de cicle de la línia de muntatge. Aquest problema es sol presentar quan la línia de muntatge ja existeix.
- c) SALBP-E: es busca maximitzar l'eficiència de la línia, és a dir, minimitzar el producte de nombre d'estacions pel temps de cicle.
- d) SALBP-F: es busca trobar una solució factible per a una combinació qualsevol de temps de cicle i nombre d'estacions.

B. GALBP: General Assembly Line Balancing Problem

Aquest problema engloba a tots els tipus de ALBP que no són el SALBP. Són exemple del GALBP, problemes com: models mixts, estacions en paral·lel, rutes de procediment alternatives, temps de procés variables,... Aquests problemes es caracteritzen per representar problemes més reals que els SALBP.

Dins de la varietat dels problemes GALBP, destaquen els següents tipus:

- a) UALBP (U-line Assembly Line Balancing Problem): en comptes de considerar línies serials, es treballa amb línies de tipus U. Becker i Scholl (2006, p.694- 715) estableixen que en una línia de tipus U, les estacions poden ser col·locades de tal manera que es poden manejar alhora dues peces en diferents posicions de la línia. En aquest cas també es poden distingir diferents variants: UALBP-1 (minimitzar el nombre d'estacions), UALBP-2 (minimitzar el temps de cicle) i UALBP-E (maximitzar l'eficiència de la línia).
- b) MALBP (Mixed-Model Assembly Line Balancing Problem): consisteix en l'equilibrat de línies amb models mixtos. Es caracteritza per tenir diversos models d'un mateix producte, amb un conjunt de tasques comunes per a tots els models, i sense tenir en compte temps de preparació.
- c) RALBP (Robotic Assembly Line Balancing Problem): consisteix en l'equilibrat de línies robotitzades. En aquest tipus de problema es vol optimitzar algun criteri de la línia

considerant l'assignació de les tasques a cada estació i tenint en compte els temps de desplaçament dels robots.

- d) MOALBP (Multi-Objective Assembly Line Balancing Problem): consisteix en l'equilibrat amb objectius múltiples. Aquests objectius poden ser, per exemple: minimitzar el nombre d'estacions, el cost o el nombre de pulmons i / o maximitzar l'eficiència. Rekiek (2002) estableix que la majoria de problemes d'equilibrat de línies de muntatge en la vida real presenten objectius múltiples.
- e) LB-ALBP (Lexicographic Bottleneck Assembly Line Balancing Problem): varietat del SALBP-2 que consisteix en que, a més de minimitzar el temps de cicle del sistema, es vol minimitzar el temps de la segona estació més carregada, el de la tercera, ... i així successivament.
- f) PALBP (Parallel Assembly Line Balancing Problem): consisteix a tenir dues o més línies de muntatge en paral·lel on els temps de cicle de cada línia poden variar i on l'objectiu principal és minimitzar el nombre d'operaris. Aquest problema té la particularitat que els operaris d'una línia poden treballar també a la línia contigua.

4. Parallel Assembly Line Balancing Problem (PALBP)

El problema s'introdueix per primera vegada per Gökçen et al. [1]. Varen proposar diversos procediments i un model matemàtic de programació per a la línia de muntatge d'un problema d'equilibri que conté línies paral·leles. L'objectiu del seu estudi va ser minimitzar el nombre d'estacions, a la vegada que s'equilibrava més d'una línia de muntatge i es permetia la unió d'estacions.

4.1. Característiques principals del problema a resoldre

Tal i com s'ha comentat a l'apartat 3.1., existeixen diferents tipus de línies per equilibrar. Seguidament es detallen les característiques principals del problema de línies en paral·lel (PALBP).

Hi ha dos subproblemes bàsics en PALBP: l'assignació de productes en línies i l'equilibratge de línies adjacents. Tradicionalment, cada línia s'equilibra per separat sense compartir operaris tenint en compte les relacions de precedència entre les tasques i el temps de cicle per tal de reduir al mínim el nombre total d'estacions de treball. Tot i això, el disseny de línies paral·leles pot oferir un menor nombre d'operaris que un conjunt de línies individuals. Si aquestes línies paral·leles s'instal·len de tal manera que un únic treballador pugui funcionar en ambdues direccions, és possible reduir el nombre d'operaris. En altres paraules, el propòsit principal de PALBP és equilibrar el muntatge de línies paral·leles simultàniament. Això s'aconsegueix mitjançant l'assignació de tasques de diverses línies veïnes a un operador polivalent. Aquest disseny amb dues línies paral·leles està representat esquemàticament a la Figura 7, on els treballadors 1 i 3 poden operar en ambdues línies conjuntes. Per poder resoldre aquests tipus de problemes es necessita saber:

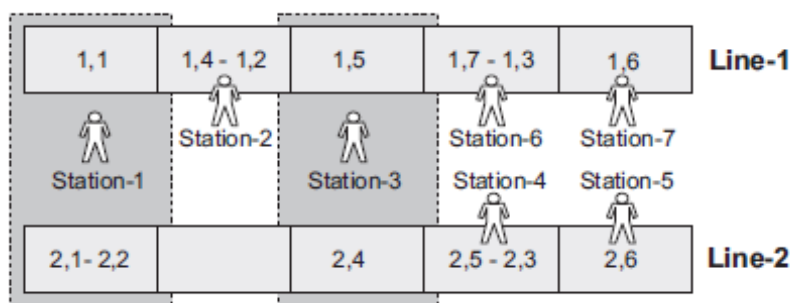


Figura 10: Representació esquemàtica del PALB. [Font: [7]]

- Per a cada producte, el nombre de tasques a processar, la seva durada (que és independent de l'operari que la realitzi) i les seves relacions de precedència, així com el temps de cicle a respectar (la productivitat a assolir).
- El nombre màxim de columnes d'estacions en totes les línies és el mateix i igual a K ($K=5$, a la Figura 7).
- Poden existir estacions intermèdies sense cap operari assignat (per exemple en l'estació 2 de la línia 2, de la Figura 7).
- Un operari es pot assignar a una estació d'una línia (per exemple a l'estació 2 de la línia 1, a la Figura 7) o a dues estacions de dues línies adjacents però del mateix ordre (per exemple a l'estació 1 de les línies 1 i 2, de la Figura 7)

Tenint en compte el que s'ha exposat anteriorment, i donat que els temps de cicle a respectar són coneguts en un principi i que tots els operaris tenen el mateix sou, l'objectiu del treball consistirà a minimitzar el número total d'operaris que són necessaris sense superar els límits físics de la instal·lació (K columnes d'estacions) o el que és el mateix, augmentar al màxim l'eficiència de la línia (LE) que vindria donada per la següent equació:

$$LE = \frac{\sum_{i=1}^R ST_i}{TC \cdot R} \cdot 100\% \quad (Eq. 1)$$

On:

R = Total nombre d'estacions utilitzades.

ST_i = Temps d'ocupació de l'estació i .

TC = Temps de cicle.

A causa de la dificultat del problema a resoldre, es treballarà amb una versió del mateix en què hi haurà dues línies de muntatge. A més a més, es proposarà una resolució del problema quan considerem que el temps de cicle és el mateix a les dues línies i quan el temps de cicle de la línia 2 és el doble que el de la línia 1.

El problema PALB té dues maneres diferents de resolució: l'equilibratge conjunt o equilibratge individual. L'equilibratge conjunt consisteix a equilibrar una sola línia, prenent com a candidates les tasques del producte A i del producte B, sempre i quan es compleixin les precedències. Aquesta única línia s'ha de desglossar en dues. Les estacions que continguin

tasques d'A i B seran compartides i les que només tinguin tasques d'un producte seran individuals. A més a més, s'haurà d'introduir estacions buides sempre que sigui necessari. Aquesta metodologia era plantejada per Gökçen en les seves propostes de resolució per programació matemàtica i heurística [1] i per la metaheurística de recerca tabú (Tabu Search) [2]. La tendència és a compartir operaris i, per tant, a adjuntar estacions de línies diferents, el qual permet acostar-se molt a una fita inferior. No obstant això, provoca que es superi amb escreixos el límit de columnes d'estacions. És per aquesta raó que per tal de complir amb el requisit d'espai límit, s'haurien de trencar comparticions i comprimir les tasques.

D'altra banda, la segona manera de resolució es basa a aplicar dos problemes d'equilibratge de línia individual (Single Assembly Line Balancing Problem - SALBP), un per a cada producte. El resultat obtingut són dues línies equilibrades de forma independent. La tendència en aquest cas és omplir al màxim les estacions amb tasques d'una única línia (no hi haurà comparticions d'operaris), de manera que amb molta probabilitat es complirà amb el requisit d'espai. Tanmateix, en poques ocasions s'aconseguirà compartir operaris en una mateixa columna d'estacions degut a que el temps restant de cada operari serà molt petit. Per tant, per aconseguir millors solucions s'hauran d'estendre les tasques en les estacions i així intentar compartir operaris.

4.2. Algoritmes per al PALBP

4.2.1. Mètodes exactes

Aquests procediments garanteixen trobar la millor solució en un temps finit. No obstant, el temps invertit pot ser d'ordre de magnitud molt elevat i per tant molt superior al del mètode no exacte (heurístic) que veurem en l'apartat 4.2.2..

Tècniques de ramificació i afitament (B&B, branch & bound)

Scholl i Boysen [5] modelen el problema com un problema binari de programació lineal i varen proposar un procediment de solució exacta a partir de l'algorisme de Branch and Bound.

En un treball posterior, Scholl i Boysen [6] van esmentar que la funció objectiu original (minimitzar el nombre d'estacions de treball) no és suficient per a l'obtenció d'un disseny d'una línia competitiva, ja que podria donar lloc a ineficiències

Branch and Bound (B&B) és una tècnica eficient per a la cerca de solucions on el seu objectiu no és trobar una solució qualsevol si no l'òptima. L'algorisme B&B interactua i crea un arbre de solucions, on cada una de les rames ens porta a una possible solució posterior a l'actual,

El funcionament general de l'algorisme B&B consisteix a reduir l'espai de cerca del problema podant les zones de l'arbre encara no explorades que no poden crear millors solucions que la solució en curs. Per tant, el mètode de ramificació i afitació B&B busca sempre la millor solució.

A continuació, es mostra un exemple de la ramificació i es veu com es poden certes “rames” en la configuració. En resum, per a cada un dels diferents nivells ens quedarem amb el valor que ens pugui aportar una solució millor i els altres els descartarem. D'aquesta manera, anirem reduint les possibles solucions fins a arribar a la millor possible. En la següent figura es veu com es parteix d'un primer valor i a partir d'aquí es creen altres valors de possibles solucions (rames), la part vermella significa que s'ha “podat” el arbre i per tant no es seguiran explorant les possibles solucions ja que la part verda té un valor millor.

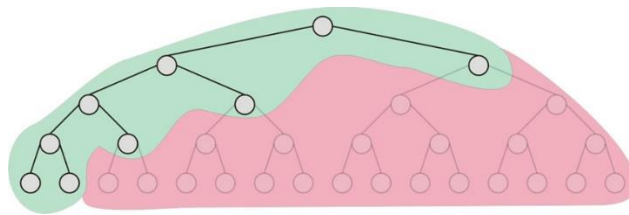


Figura 11: Exemple de ramificació del mètode Branch and Bound. [FONT: control.isy.liu]

4.2.2. Mètodes no exactes

Tal i com s'ha comentat en l'apartat anterior, els mètodes exactes poden tenir un temps d'execució molt elevat, és per això que s'expliquen algunes de les metaheurístiques més populars

El concepte metaheurística engloba aquelles tècniques basades en la interacció de procediments d'exploració de solucions veïnes que permetin al procés fugir dels òptims locals i realitzar una exploració robusta de l'espai de solucions.

El gran avantatge de les metaheurístiques és la seva versatilitat: el fet d'utilitzar conceptes estratègics independents del problema permet que es pugui aplicar a un gran ventall de problemes d'optimització. És a dir, partint d'un esquema metaheurístic general, es pot adaptar al problema que s'estigui resolent. Alguns mètodes de resolució no exactes estan basats en principis útils per a la resolució de problemes, com pot ser la cerca de millors solucions del veïnat. En canvi, unes altres simulen fenòmens naturals, com pot ser l'evolució genètica que es produeix en la reproducció o la “comunicació entre formigues”.

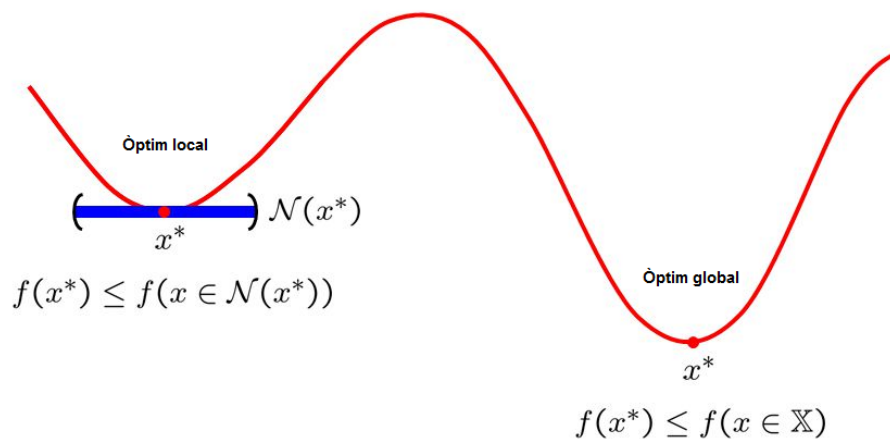


Figura 12: Diferenciació entre un òptim local i un òptim global. [FONT: pròpia]

Simulated Annealing (SA)

Quan parlem del Simulated Annealing ens referim al procés de temperat dels metalls que s'utilitzava aproximadament als anys 5000 a.C. Aquest procés està caracteritzat per la presència de tres fases diferenciades: en primer lloc, la fase d'escalfament del metall a una temperatura preestablerta; la segona fase es manté aquesta alta temperatura per tal que les molècules puguin acomodar-se en estats de mínima energia i finalment, l'última fase que consisteix a aplicar un procés de refredament controlat per tal d'augmentar el volum dels cristalls i reduir-ne els seus efectes.

En aquesta línia, cal destacar l'algorisme de Metrópolis [10] desenvolupat l'any 1953, el qual ha esdevingut el pioner en mètodes de Simulated Annealing. Si més no, cal considerar també les aportacions realitzades per Kirkpatrick i Gelatt [11], els quals van ser els primers en aplicar-ho als problemes d'optimització per tal de trobar solucions al problema de venedor viatger amb un número relativament gran de ciutats.

Així doncs, la metaheurística del Simulated Annealing (SA) el podem descriure com un mètode iteratiu que s'inicia amb un cert estat s . A través d'un procés particular genera un estat veí s' a l'estat actual. Si l'energia, o avaluació, de l'estat s' és menor que la de l'estat s , es canvia l'estat s per s' .

En cas que l'avaluació de s' sigui major que la de s , es pot empitjorar escollint s' en lloc de s amb una certa probabilitat que depèn de les diferències de les avaluacions $\Delta f = f(s) - f(s')$ i de temperatura actual del sistema T . La possibilitat de triar un estat pitjor a l'actual és el

Simulated Annealing pot passar dels òptims locals i poder arribar als òptims globals. La probabilitat d'acceptar la tria d'un pitjor estat, normalment es calcula a través del criteri de metròpoli que trobem a continuació:

$$P(\Delta f, T) = e^{\Delta f / T} \quad (Eq. 2)$$

En el Simulated Annealing la temperatura va decreixent iterativament de tal manera que cada cop hi haurà menys probabilitat que s'accepti una solució pitjor a l'actual.

Tabú Search (TS)

Ozcan [2] va desenvolupar un algorisme Tabú per resoldre PALBP amb l'objectiu de maximitzar l'eficiència de la línia i la minimització de la variació de les càrregues de treball. A més a més és tracta del primer article en tenir més d'un objectiu per la resolució del problema.

És un algorisme basat en l'ús de la memòria per guiar el procediment de cerca de l'algorisme. Fa ús d'una memòria a curt termini, que manté una llista tabú, que impedeix a la cerca tornar a solucions ja visitades.

En una memòria a curt termini, es guarda el llistat de propietats que estan restringits per a l'algorisme (llista tabú). Els elements es mantenen en la llista durant cert nombre d'iteracions fixat per un paràmetre. Normalment aquesta està formada per les últimes transformacions de la solució que eviten moviments cíclics que retornin la cerca a solucions ja visitades. Per a alguns problemes és recomanable guardar i mantenir diferents llistes tabú o utilitzar llistes de grandària variable.

Per tal de decidir si s'accepta una d'aquestes solucions tabú, s'han de complir un criteri d'aspiració. El compliment d'aquest significa que la solució generada en aquest instant no és tabú, fins i tot si el moviment que la produeix l'és. Es defineixen perquè les llistes tabús poden ser contraproductives, ja que pot donar-se el cas que prohibeixin moviments atractius o que, simplement, produeixin estancament. El criteri d'aspiració més usat és admetre solucions si aquesta és millor que la millor obtinguda fins al moment.

Greedy Randomized Adaptive Search Procedure (GRASP)

GRASP és un procediment iteratiu on cada iteració consisteix en una fase de construcció i una fase de millora.

La fase constructiva construeix iterativament una solució factible, afegint un element a cada pas. A cada iteració, l'elecció de la següent tasca per ser afegida a la solució parcial s'escull a l'atzar entre una llista d'elements mitjançant un indicador heurístic. És necessari ressaltar que aquesta elecció no té en compte en cap moment que passarà en iteracions futures.

Es a partir d'una primera solució factible on comença la fase de millora. Aquesta fase consisteix normalment en realitzar una cerca local de solucions veïnes. L'efectivitat de l'exploració depèn en gran mesura de la fase anterior, a causa de la solució de partida, encara que també cal tenir en compte l'estructura del veïnat i el mètode de cerca.

Genetic Algorithms (GA)

En un algorisme genètic es pretén resoldre el problema mitjançant la selecció natural. Selecció dels individus més aptes i encreuament per generar descendents millor adaptats, tenint en compte que es poden produir mutacions durant el procés.

El procediment consisteix a seleccionar solucions d'una població de solucions i combinar-les segons un operador d'encreuament, de manera que es millori la població de solucions. S'afegeix la possibilitat de modificar aleatòriament les solucions, és a dir, es produeixen mutacions

Un algorisme genètic pot presentar diverses variacions, depenent de com s'apliquen els operadors genètics (creuament, mutació), de com es realitza la selecció i de com es decideix el reemplaçament dels individus per formar la nova població. En general, l'algorisme consisteix en els següents passos:

- **Inicialització:** Es genera aleatòriament la població inicial, que està constituïda per un conjunt de cromosomes els quals representen les possibles solucions del problema. En cas de no fer-ho aleatòriament, és important garantir que dins de la població inicial, es tingui la diversitat estructural d'aquestes solucions per tenir una representació de la major part de l'espai de solucions o almenys evitar la convergència prematura.
- **Avaluació:** A cadascun dels cromosomes d'aquesta població s'aplicarà la funció d'aptitud per saber com de "bona" és la solució que s'està codificant.
- **Condició de terme:** Normalment s'utilitzen dos criteris d'aturada: realitzar el algorisme un nombre màxim d'iteracions (generacions) o detenir-lo quan no hi hagi canvis en la població. Mentre no es compleixi la condició de terme es fa el següent:
 - ✓ **Selecció:** Després de saber l'aptitud de cada cromosoma es procedeix a triar els cromosomes que seran creuats en la següent generació. Els cromosomes amb millor aptitud tenen major probabilitat de ser seleccionats.

- ✓ **Recombinació o Creuament:** La recombinació és el principal operador genètic, representa la reproducció sexual, opera sobre dos cromosomes alhora per generar dos descendents on es combinen les característiques de tots dos cromosomes progenitors.
- ✓ **Mutació:** Modifica a l'atzar part del cromosoma dels individus, i permet aconseguir zones de l'espai de cerca que no estaven cobertes pels individus de la població actual.
- ✓ **Reemplaçament:** Una vegada aplicats els operadors genètics, es seleccionen els individus (normalment els millors) per conformar la població de la generació següent.

Ant Colony Optimization (ACO)

Ozbakir et al. [4] van desenvolupar un dels primers intents de solució a partir d'un nou algorisme de les formigues de diverses colònies, desenvolupat per a l'equilibri de línies de muntatge en paral·lel basat en metaheurístiques.

Els algorismes d'optimització mitjançant colònia de formigues es caracteritzen per basar-se en el comportament de les formigues quan formen les rutes entre el niu i la font d'aliment, sobre la base d'un rastre de feromones que dipositen en la seva trajectòria efectuada.

Són procediments iteratius on cadascuna de les formigues de la colònia genera una solució en cada iteració, i aquestes es succeeixen fins a aconseguir una condició de finalització com en les altres metaheurístiques. També és necessari afegir a l'algorisme l'actualització del rastre de feromones en cada iteració per poder mantenir la informació obtinguda amb les solucions generades.

L'elecció és aleatòria, a cada component se li assigna una probabilitat de ser escollit i es basa en l'experiència acumulada amb les solucions anteriors. La probabilitat és una funció que depèn del rastre artificial de feromones a més a més d'alguns components heurístics com podria ser un indicador i simula el comportament de les formigues en la naturalesa en l'elecció de les rutes (veure Equació 3).

$$p_{ij}^h = \begin{cases} \frac{[\tau_{ij}]^\alpha}{\sum_{l \in N_i^h} [\tau_{il}]^\alpha} & \text{si } j \in \delta_i^h \\ 0 & \text{si } j \notin \delta_i^h \end{cases} \quad (\text{Eq. 3})$$

On p_{ij}^h és la probabilitat que la formiga h esculli el component j en la decisió i , τ_{ij} representa la intensitat del rastre de feromones en l'opció j de la decisió i , α és un paràmetre i δ_i^h és el conjunt de components que es poden afegir en la decisió i de la formiga h .

Finalment, els algorismes ACO també poden incorporar un procés de cerca local al procediment per a la millora de les solucions generades.

5. Resolució del PALBP

5.1. Elements comuns

Primer de tot cal dir que aquest dos procediments de resolució que es dissenyen i s'implementen no han estat proposats per cap altre autor. Seguidament és mostra com calcular la fita mínima en tots dos procediments.

5.1.1. Fita mínima

Per a les solucions obtingudes de procediments de resolució no exactes, com ara les heurístiques o les metaheurístiques, és bo tenir un valor de referència anomenat fita mínima. Aquest valor ens marca el valor pel qual per sota serà impossible trobar una solució factible. Per aconseguir aquest valor de referència i poder jutjar la qualitat de la solució obtinguda, cal relaxar el problema. Per fer-ho, es suposa que les tasques es poden dividir i per tant poden estar-se realitzant a més d'una estació. El menor nombre d'operaris possible en aquest cas ve donat per l'expressió que trobem a continuació:

$$C_{min} = \left\lceil \frac{\sum_{j \in \{A,B\}} (\sum_{i=1}^{n_j} t_{i,j})}{TC} \right\rceil \quad (Eq. 4)$$

Aquesta fita mínima, no ha de ser l'òptima del problema, ja que s'ha obtingut relaxant aquest. És a dir, podria donar-se el cas que en un enunciat fos impossible arribar a aquesta fita tot i haver trobat la millor solució possible. La fita ens ajuda a tenir una referència i en algunes ocasions permet trobar la solució més ràpidament ja que si es troba una solució amb un valor igual a la fita, significarà que ja no podem trobar una solució millor.

5.2. Heurística Ad hoc

5.2.1. Funcionament de l'heurística Ad hoc

Quan parlem de les cerques locals, cal que tinguem present que el seu principal inconvenient és que aquestes presenten solucions localment òptimes que en moltes ocasions poden estar molt allunyades de la solució òptima global.

A la Figura 16 podem veure l'esquema general de l'heurística Ad hoc que s'ha proposat. Podem descriure dues fases en cada iteració i . En el primer cas, es construeix una solució x_i , mentre que la segona es tracta de millorar-la, mitjançant l'aplicació d'un mètode de cerca, obtenint la solució x'_i (que ocasionalment pot ser igual a x_i).

```

Inicialitzar  $no = 2 \cdot \bar{K} + 1$ 
Mentre (Condicció d'aturada)
     $no = no - 1$ 
    Fase 1 Generar solució inicial  $x_i$ 
    Fase 2 Millorar la solució  $x_i$ 
         $x'_i$  és la solució millorada
    Actualitzar la millor solució a  $x'_i$ 
Retornar
  
```

Figura 13: Pseudocodi. [FONT: pròpia].

5.2.2. Heurística Ad hoc

En aquesta resolució el que es fa és crear una configuració d'operaris per a totes les estacions. Una configuració consisteix en decidir per a cada columna d'estacions si hi haurà dos operaris (representat amb la lletra B) o un únic operari que realitzarà tasques tant del producte A com del producte B (representat amb la lletra S).

A partir d'aquí es resol mitjançant una heurística *greedy*, la qual selecciona la millor candidata factible d'una llista ordenada segons un indicador. El procediment que es segueix en aquesta heurística es el següent:

1. Partint de $no = 2 \cdot \bar{K} + 1$. El valor inicial del numero d'operaris és el màxim permès ($2 \cdot \bar{K}$) més una unitat, això permet que quan el bucle comenci, es col·loqui amb el màxim possible ja que com es veurà en el següent pas, a cada iteració es resta un operari.
2. Creació d'una configuració per a totes les estacions on $no = no - 1$ en cas de que sigui la primera iteració o $no = \text{millor solució operaris}$. El valors d'aquesta configuració poden adquirir el valor B (un operari a cada línia en aquella estació) o S (un operari per a les dues línies en aquella estació).

La configuració en el primer cas seran tot B, ja que el nombre d'operaris és el màxim possible. A partir de la segona iteració, el nombre de $B = 2 * (\bar{K} - (2 * \bar{K} - no))$, d'aquesta manera, obtindrem que la resta d'estacions seran S. Aquesta configuració es podria fer de diferents formes, com per exemple considerant la possibilitat que a una estació només s'assignessin tasques a la línia 1 o la línia 2 o be cap tasca. Fent diferents proves s'ha observat que les solucions mai són millors que les generades amb una configuració on només hi ha S o B. El fet que la configuració sigui aleatòria pot fer que variï la solució final depenent d'on es col·loquin les B i les S. És per això, que es generen constantment diferents solucions en cas d'empitjorar l'última solució trobada.

3. Càlcul del valor de l'indicador per a cada tasca i del producte j . Cal esmentar que les heurístiques *greedy* són deterministes, és a dir, proporcionen una única solució per a cada indicador. És per aquesta raó que es decideix definir una sèrie d'indicadors diferents per obtenir un conjunt de solucions factibles. Els indicadors [8] utilitzats per a cada producte són:

INDICADOR	DESCRIPCIÓ	EXPRESSIÓ INDICADOR
1	Ordenar en ordre no decreixent segons el temps de procés de la tasca i del producte j .	$t_{i,j}$
2	Ordenar en ordre decreixent segons el temps de procés de la tasca i del producte j .	$t_{i,j}$
3	Ordenar en ordre no decreixent segons el nombre de successors immediats de la tasca i .	NIS_i
4	Ordenar en ordre no decreixent segons el nombre de successors totals de la tasca i .	NS_i
5	Ordenar en ordre no decreixent segons el temps de procés de la tasca i més tots els temps de procés de les seves tasques successors totals.	$t_{i,j} + \sum_{m \in S_i} t_{m,j}$
6	Ordenar en ordre no decreixent segons el temps mig de les tasques restants de la tasca i .	$\frac{t_{i,j} + \sum_{m \in S_i} t_{m,j}}{NS_i + 1}$

7	Ordenar en ordre no decreixent segons l'estació earliest ($-E_i$) de la tasca i .	$E_i = \left\lceil \frac{t_{i,j} + \sum_{m \in P_i} t_{m,j}}{TC} \right\rceil$
8	Ordenar en ordre no decreixent segons l'estació <i>lastest</i> ($-L_i$) de la tasca i .	$L_i = \bar{K} + 1 - \left\lceil \frac{t_{i,j} + \sum_{m \in S_i} t_{m,j}}{TC} \right\rceil$
9	Ordenar en ordre no decreixent segons el rang en que pot començar la tasca i .	$-(L_i - E_i)$
10	Ordenar en ordre no decreixent un indicador que considera: el temps de la tasca i i el nombre de successors immediats (paràmetres a calibrar).	$\alpha \cdot t_{i,j} + \beta \cdot NIS_i$
11	Ordenar en ordre no decreixent un segon indicador que considera: el temps de la tasca i i el nombre de successors totals (paràmetres a calibrar).	$\gamma \cdot t_{i,j} + \sigma \cdot NS_i$
12	Ordenar en ordre no decreixent un tercer indicador que considera: el temps de procés de la tasca i més tots els temps de procés de les seves tasques successors totals (paràmetres a calibrar).	$\mu \cdot t_{i,j} + \theta \cdot \sum_{m \in S_i} t_{m,j}$

Taula 1: Descripció i expressions matemàtiques de cadascun dels indicadors utilitzats. [FONT: pròpia]

Els indicadors 10, 11 i 12 contenen paràmetres que s'han de calibrar. Calibrar consisteix en fixar els valors dels diferents paràmetres. En aquest cas es calibrarà amb la utilització d'un algoritme directe i utilitzant un conjunt d'exemplars de calibratge. S'utilitza un software anomenat CALIBRA (Adenso-Díaz i Laguna, 2006), que permet utilitzar conjuntament el disseny d'experiments factorial de Taguchi i de recerca local. Per tal de millorar el calibratge, s'utilitzarà el programa dues vegades seguides (CALIBRA + CALIBRA). El fer d'executar el programa dues vegades, ens permet afitar molt més el valors retornats en la primera fase. Per calibrar els paràmetres s'han utilitzats 40 exemplars de calibratge diferents. Un cop executat el programa CALIBRA, s'ha vist que els millors valors són els següents:

PARÀMETRES	VALOR	PARÀMETRES	VALOR
α	1.1	σ	10.5
β	4.8	μ	5.1
γ	0.9	θ	1.2

Taula 2: Paràmetres utilitzats per als indicadors 10,11 i 12. [Font: pròpia]

4. S'inicia el comptador de columnes d'estacions on podem tenir dues B o una S. En cas de que el comptador sigui més gran que \bar{K} , significarà que ja no podem posar més tasques i haurèm de passar a fer la reparació de la solució.
5. Tenint en compte les precedències, es crea o s'actualitza la llista de candidates.
6. És selecciona la tasca i que estigui en la posició més preferent i s'introdueix en l'estació k . No obstant això, si el temps de procés de la tasca i més el temps de les tasques incloses en l'estació pertinent supera el temps de cicle, es comprova la tasca $i+1$, i així successivament fins que cap tasca es pugui introduir.

En el cas de tenir una B, s'estudien les dues línies per separat cadascuna amb les seves candidates. No s'augmenta el comptador d'estacions fins que totes dues línies han arribat al màxim d'ocupació de tasques. En el cas de tenir una S, es tracta com una estació única on el conjunt de candidates són tant del producte A com del producte B i es tria la millor segons cada indicador.

7. Si la llista de candidates està buida, passarem al següent pas, el número 8. En canvi, si encara hi ha tasques a la llista però ja tenim plenes les estacions de les dues línies, augmentarem el comptador d'estacions en 1 unitat i anirem al pas 3.

Una vegada comprovada la llista de candidates, en cas de no tenir cap tasca com a candidata significarà que totes les tasques han estat assignades i per tant es pot passar al pas 8. D'altra banda, si queden tasques per assignar, es passa a la següent estació sempre i quan no es superi el límit d'estacions marcat per \bar{K} . En cas de superar aquest límit, s'haurà de passar a la reparació de la solució (apartat 5.2.3).

8. Totes les tasques han estat assignades a una estació. En aquest punt, cal calcular el nombre d'operaris utilitzats i en cas de ser un valor igual o més petit al cercat anteriorment, actualitzem *millor solució operaris* = no_{real} . Finalment, tornarem al pas 2 per tal de trobar una solució amb menys operaris.

Una vegada s'arriba a aquest pas significa que totes les tasques han estat assignades i per tant, ja es té una solució factible. És en aquest punt, on es recalcula el nombre d'operaris utilitzats, ja que tot i tenir feta una configuració, pot ser que hi hagi operaris sense tasques. Un cop trobat el valor del nombre d'operaris reals, en cas de ser millor que el valor que es tenia, s'actualitza. Amb aquest valor de nombre d'operaris es retrocedeix fins al pas 2 i s'intenta trobar una solució factible amb un operari menys que la millor solució factible.

Cal afegir que per arribar a aquest procediment, s'han provat diferents variants. A més a més de l'opció de ocupar només una estació de la línia 1 o una estació de la línia 2, es va provar el fet de forçar que hi hagués estacions buides. S'ha vist per tant que és millor ocupar el màxim d'estacions possibles sempre i quan no es superi el màxim (K). A continuació es mostra dues configuracions on en una d'elles hi ha una columna d'estacions buides i en l'altra no. S'ha comprovat que la segona configuració ofereix igual o millors solucions que la primera, però en cap cas pitjors solucions.



Figura 14: Configuració 1 permeten tenir estacions buides. [FONT: pròpia]



Figura 15: Configuració 2 obligant que totes les estacions estiguin plenes. [FONT: pròpia]

5.2.3. Reparació de la solució

Aquesta reparació de la solució podria ser que mai es dugués a terme. Com s'ha explicat en l'anterior apartat, en cas que no es puguin assignar totes les tasques, es realitzaria aquesta reparació per tal d'intentar trobar una solució factible. A causa del fet que en la primera solució la configuració estarà formada únicament per "B" (màxim d'operaris possible) ens assegurarem de trobar una primera solució per tant no es realitzaria aquest apartat.

En cas de trobar la fita mínima i no tenir cap tasca sense assignar, significarà que s'ha trobat la millor solució possible i per tant no caldrà realitzar aquest apartat. D'altra banda, en cas de tenir tasques pendents es passaria a fer la següent reparació:

Aquesta reparació consisteix a intentar modificar les tasques d'estacions per poder inserir quan es pugui les tasques no assignades fins ara. Per fer-ho, s'utilitza un procés iteratiu on la condició de fi ve marcada per una condició d'assignació que es veurà seguidament.

1. El primer pas d'aquesta optimització serà intentar col·locar una tasca no assignada a una de les estacions complint les precedències. D'aquesta manera, es provarà tant amb les tasques no assignades de la línia 1 com de la línia 2, i es provaran de col·locar en totes les estacions.

En la primera iteració, mai trobarà una tasca no assignada que es pugui assignar a una estació. El que sí que pot esdevenir és que a mesura que es facin les iteracions i es vagin descol·locant les tasques, una estació disposi de més temps lliure que a l'inici i per tant es pugui col·locar allí una tasca.

2. El segon pas, el qual només es realitzarà si no s'ha pogut fer el primer pas, consisteix a intercanviar una tasca no assignada per una que ja ha estat assignada, respectant sempre les respectives precedències. De la mateixa manera que passava en el primer pas, es provarà per totes dues línies i per totes les tasques. El fet de jugar amb diferents indicadors pot desembocar en què la primera iteració sigui factible realitzar aquest canvi i que compleixi la condició d'assignació que s'explicarà posteriorment i per tant es produeixi el canvi.
3. El tercer pas, el qual es realitzarà si no s'ha pogut fer ni el primer ni el segon, es tracta de col·locar una tasca que ja està assignada a una altra estació diferent, sempre respectant les precedències i fent-ho per totes dues línies i totes les tasques i estacions possibles. Aquest pas permetrà buidar més una estació i poder col·locar més endavant alguna tasca que no hagi estat assignada fins ara. Per tal de poder realitzar correctament aquest pas, s'haurà de complir la condició d'assignació que s'explicarà posteriorment.
4. L'últim pas, el qual es produirà en cas de no haver trobat cap solució en els altres tres casos, consisteix en intercanviar dues tasques que ja han estat assignades d'una estació a una altra. Es provaran les dues línies i s'han de complir tant el temps de cicle com el llistat de precedències. Aquest intercanvi s'intentarà per totes les tasques tenint en compte que no poden estar en una mateixa estació. D'igual manera que en els

altres passos, s'haurà de complir una condició d'assignació que s'explica a continuació.

La condició de fi vindrà marcada per la condició d'assignació. És a dir, en cas de que s'arribi al final del punt 4 explicat anteriorment i la condició d'assignació no sigui certa, és sortirà del bucle iteratiu.

La primera condició es la següent:

$$\sum_{p \in P} \left(\sum_{(p,j) \in UT(\Delta)} t_{pj} \right)^2 \leq \sum_{p \in P} \left(\sum_{(p,j) \in UT(\Delta')} t_{pj} \right)^2 \quad (Eq. 5)$$

En aquesta primera condició el que es comparà és el quadrat del temps de les tasques que no han estat assignades ($UT(\Delta)$), i es compara la solució que ja es té $UT(\Delta')$ amb la possible solució $UT(\Delta)$. Aquesta primera condició d'assignació només és útil en el segon cas, ja que en el primer cas, sempre que es puguin treure tasques no assignades, beneficiarà a l'hora de trobar una solució factible. En els dos últims casos, no es toquen les tasques no assignades, per tant la igualtat anterior serà la mateixa en els dos costats.

-Exemple de la primera condició: Suposem el cas que en la primera solució tenim un conjunt de tasques sense assignar de la línia 1, en la qual els seus temps són 2,3 i 4. Per altra banda, a la línia 2, tenim tasques sense assignar que els seus temps són 4 i 6.

En la nova solució s'intercanvia una tasca que ja estava assignada de temps 3 per una tasca no assignada de la línia 1 de temps 4. D'aquesta manera la equació 4 quedaria:

$$(2 + 3 + 3)^2 + (4 + 6)^2 \leq (2 + 3 + 4)^2 + (4 + 6)^2 \Rightarrow 164 < 181 \Rightarrow CERT$$

En aquest cas s'acceptaria la solució com a millor i s'actualitzarien els diferents vectors de tasques assignades i no assignades.

La segona condició d'assignació la qual s'estudiarà en cas d'empat en la primera condició és la següent:

$$\sum_{(h,k) \in K} \sqrt{\sum_{(p,h,j) \in T_k} t_{hp,j}} < \sum_{(h,k) \in K} \sqrt{\sum_{(p,h,j) \in T'_k} t_{hp,j}} \quad (Eq. 6)$$

En aquesta segona condició el que es mira és l'arrel del temps total de les tasques assignades per estació (suma de línia 1 i línia 2). D'aquesta manera el que s'intenta és tenir un major

desequilibri en una estació per fer que sigui més probable col·locar una nova tasca. És per això, que es compara l'assignació de tasques trobades en la solució que ja es té T'_k amb la possible solució T_k .

-Exemple de la segona condició: Suposem el cas que en la primera solució tenim un conjunt de tasques assignades a la línia 1, en què els seus temps per a cada estació són 6 en la primera estació de la línia 1, 9 en la segona estació de la línia 1 i 4 en la tercera estació de la línia 1. Per altra banda, a la línia 2 el conjunt de tasques assignades a cada estació són 7 en la primera estació de la línia 2, 7 en la segona estació de la línia 2 i 7 en la tercera estació de la línia 2.

En la nova solució s'intercanvia una tasca de la estació 1 de la línia 1 de durada 1 per una tasca de l'estació 3 de la línia 1 de durada 4. D'aquesta manera la equació 5 quedaria:

$$\sqrt{9+7} + \sqrt{9+7} + \sqrt{1+7} < \sqrt{6+7} + \sqrt{9+7} + \sqrt{4+7} \Rightarrow 10,83 < 10,92 \Rightarrow CERT$$

En aquest cas s'acceptaria la solució com a millor i s'actualitzaria el vector de posicions de les tasques assignades.

5.2.4. Exemple de resolució de la heurística Ad hoc

A continuació es mostra un exemple de resolució on s'aniran veient els diferents punts explicats anteriorment. Seguidament es mostren les tasques i les seves precedències tant de la línia 1 com de la línia 2.

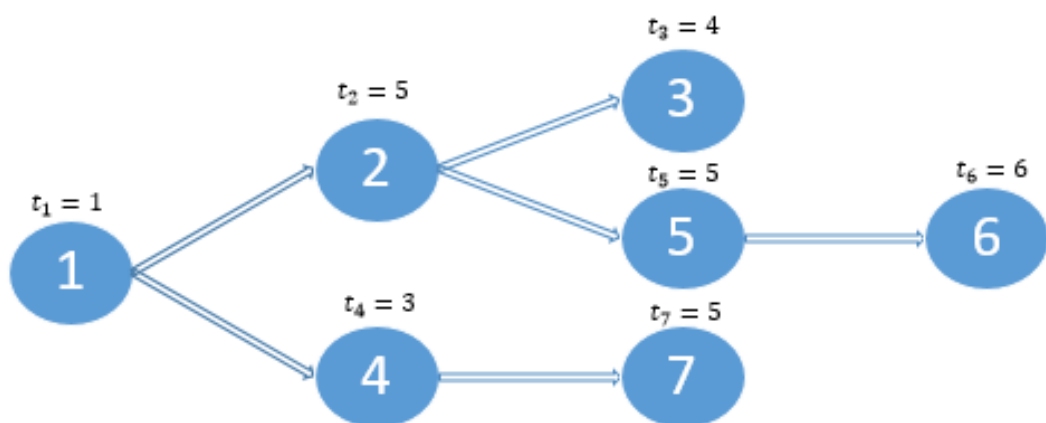


Figura 16: Graf de tasques i temps d'aquestes del producte A. [FONT: pròpia]

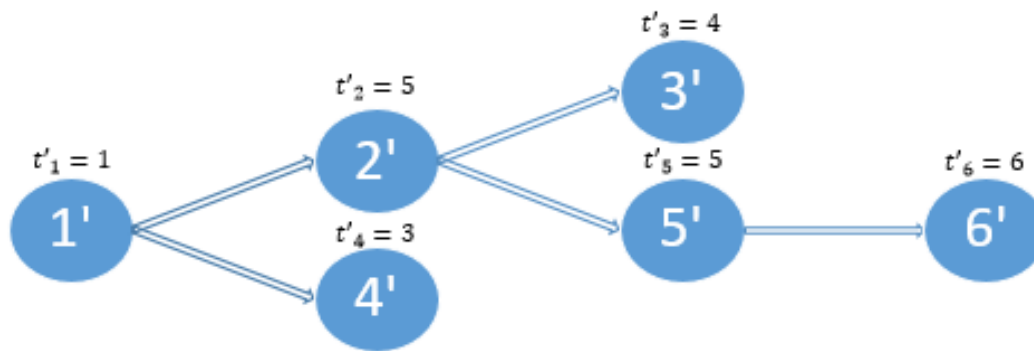


Figura 17: Graf de tasques i temps d'aquestes del producte B. [FONT: pròpia]

En aquest cas el temps de cicle (TC) per a les dues línies és de 9 segons i el nombre d'estacions K=4.

D'altra banda podem trobar el valor de la fita mínima per saber quin serà el mínim d'operaris que hauran de tenir aquestes dues línies:

$$C_{min} = \left\lceil \frac{\sum_{j=A}^B (\sum_{i=1}^{n_j} t_{i,j})}{TC} \right\rceil = \left\lceil \frac{(1 + 5 + 4 + 3 + 3 + 6 + 5) + (1 + 5 + 4 + 3 + 3 + 6)}{9} \right\rceil = 6 \quad (Eq. 7)$$

Amb aquest valor de $C_{min} = 6$, ja sabem que el nombre d'operaris estarà entre 6 i 8 ($2 \cdot K$). El fet de trobar aquest número ens pot reduir molt el temps d'execució del procediment de resolució, ja que en cas de que es trobi una solució igual al valor aquest de la fita mínima, automàticament es podrà dir que s'ha trobat la millor solució factible.

Inicialment s'ha de trobar una configuració d'operaris col·locant una "B" o una "S" a les diferents estacions. Com al principi el nombre d'operaris és el màxim possible (8) tenim que la configuració inicial serà tot de "B" tal com es veu en la imatge següent:



Figura 18: Col·locació dels operaris en la primera iteració del programa. [FONT: pròpia]

A continuació es resoldrà l'exemple amb l'indicador número 6, ja que és un dels més complexos i el procediment per als altres indicadors és molt similar.

Partint de la fórmula que s'observa en la taula 1 de l'indicador número 6, creem la següent taula de valors per a cadascuna de les tasques:

Línia 1		Línia 2	
Tasca	Valor	Tasca	Valor
1	4,14	1	4
2	5	2	5
3	4	3	4
4	4	4	3
5	5,5	5	5,5
6	6	6	6
7	5		

Taula 3: Valors obtinguts per l'indicador número 6. [FONT: pròpia]

A continuació es mostra un exemple de com trobar el valor de la tasca 1 de la línia 1:

$$\frac{t_{i,j} + \sum_{m \in S_i} t_{m,j}}{NS_i + 1} = \frac{1 + (5 + 4 + 3 + 5 + 6 + 5)}{6 + 1} = 4,14 \quad (Eq. 8)$$

El següent pas és assignar les tasques tenint en compte els valors de la taula 3 i les relacions de precedència. D'aquesta manera s'obté:

Línia 1				
Operari	Candidates	Tasca Realitzada	Temps tasca	Temps Restant
1	1	1	1	8
	2,4	2	5	3
	3,4,5	4	3	0

2	3,5,7	5	5	4
	3,6,7	3	4	0
3	6,7	6	6	3
	7	-	-	3
4	7	7	5	4

Taula 4: Assignació de tasques de la línia 1 respecte l'indicador 6. [FONT: pròpia]

Línia 2				
Operari	Candidates	Tasca Realitzada	Temps tasca	Temps Restant
1'	1	1	1	8
	2,4	2	5	3
	3,4,5	4	3	0
2'	3,5	5	5	4
	3,6	3	4	0
3'	6	6	6	3

Taula 5: Assignació de tasques de la línia 2 respecte l'indicador 6. [FONT: pròpia]

Cal remarcar, que tot i que la configuració dels operaris estigui feta per 8 operaris, la solució que es troba és de 7 operaris, consegüentment, el pas següent serà intentar trobar una configuració possible per a 6 operaris.

Primer de tot, cal trobar el nombre de "B" que hi haurà en aquesta segona iteració:

$$B = \bar{K} - (2 * \bar{K} - no) = 2 * (4 - (2 * 4 - 6)) = 4 \quad (Eq. 9)$$

Segons aquesta fórmula, el nombre de "S" serà de 2 (cada S equival a l'estació de la línia 1 mes l'estació 2 la línia 2). Una vegada disposem dels valors, cal descobrir a quines posicions es trobarà "B" i a quina "S". Aquestes, són posicions aleatòries i en l'exemple la configuració és la següent:



Figura 19: Col·locació dels operaris en la segona iteració del programa. [FONT: pròpia]

A partir d'aquí, realitzem l'assignació de les tasques. La resolució en aquest exemple es farà amb l'indicador número 6, d'igual manera que en l'anterior resolució i l'assignació de tasques a operaris serà la següent:

En aquest cas, com les tasques es comparteixen entre línies, les tasques de la línia 2 inclouran el símbol [''] al final de cada tasca.

Línia 1				
Operari	Candidates	Tasca Realitzada	Temps tasca	Temps Restant
1	1	1	1	8
	2,4	2	5	3
	3,4,5	4	3	0
2	3,5,7,3',5'	5	5	4
	3,6,7,3',5'	3	4	0
3	6,7,3',5'	6	6	3
	7,3',5'	-	-	3
4	7	7	5	4

Taula 6: Assignació de tasques de la línia 1 respecte l'indicador 6. [FONT: pròpia]

Línia 2				
Operari	Candidates	Tasca Realitzada	Temps tasca	Temps Restant
1'	1'	1'	1	8
	2',4'	2'	5	3

	3',4',5'	4'	3	0
2	3,5,7,3',5'	5	5	4
	3,6,7,3',5'	3	4	0
3	6,7,3',5'	6	6	3
	7,3',5'	-	-	3
4'	3',5'	5	5	4
	3',6'	3	4	0

Taula 7: Assignació de tasques de la línia 2 respecte l'indicador 6. [FONT: pròpia]

En aquest cas, veiem que la tasca 6' (tasca 6 de la línia 2) es queda sense assignar. Això es s'explica ja que en les estacions 2 i 3, on hi ha col·locades les dues "S", únicament s'assignen tasques de la línia 1, i per tant, les tasques de la línia 2 no s'acabaran realitzant.

En aquest punt, doncs cal passar a fer la reparació per intentar assignar la tasca 6' a una de les estacions. Els passos per fer-ho seran els següents:

Pas 1) Intentar col·locar la tasca 6' a una de les estacions. S'observa que es impossible, ja que la tasca té una durada de 6s i el valor màxim de temps mort de la línia 2 és de 3.

Pas 2) Intentar intercanviar la tasca 6' per una tasca ja assignada complint les relacions de precedència i la condició d'assignació. La tasca 6' només es podria col·locar a l'estació 4 intercanviant-la per la tasca 3. Es complirien les precedències, però el temps total seria d'11 segons que es superior al temps de cicle, per tant cal passar al pas 3.

Pas 3) Consisteix en col·locar una tasca ja assignada a una altra estació complint tant les condicions d'assignació com la relació de precedències. S'ha d'intentar fer el canvi tant en la línia 1 com en la 2, provant totes les tasques començant des de la 1.

En aquest cas, observem que quan s'arriba a la tasca 3, aquesta pot anar a l'estació 4, ja que el temps de tasca és de 4s que coincideix amb el temps restant de l'estació 4. A més a més, com que la tasca 3 no té successores, ens assegurem que es compleixen les relacions de precedència. Ara queda comprovar la condició d'assignació.

$$\sum_{p \in P} \left(\sum_{(p,j) \in UT(\Delta)} t_{pj} \right)^2 \leq \sum_{p \in P} \left(\sum_{(p,j) \in UT(\Delta')} t_{pj} \right)^2 = 0^2 + 6^2 < 0^2 + 6^2 \quad (Eq. 10)$$

Com la primera condició ens dona el mateix resultat, s'ha de passar a la segona condició d'assignació:

$$\begin{aligned} \sum_{(h,k) \in K} \sqrt{\sum_{(p,h,j) \in T_k} t_{p,h,j}} &< \sum_{(h,k) \in K} \sqrt{\sum_{(p,h,j) \in T'_k} t_{p,h,j}} = \sqrt{9+9} + \sqrt{5+5} + \sqrt{6+6} + \sqrt{9+9} \\ &< \sqrt{9+9} + \sqrt{9+9} + \sqrt{6+6} + \sqrt{5+9} = 15,11 < 15,69 \end{aligned} \quad (Eq. 11)$$

La segona equació es compleix i per tant es produiria el canvi de tasques.

A partir d'aquí es tornaria a anar al pas 1 per intentar assignar la tasca 6'. Realitzant el procediment diferents vegades, es veu que és impossible assignar la tasca 6' a cap estació, i per tant, la millor solució factible és de 7 operaris.

5.3. Simulated Annealing (SA)

A continuació s'explica la resolució basada amb la metaheurística simulated annealing explicada anteriorment.

Una de les peculiaritats d'aquest procediment es que en la millora de la solució s'explora més d'un veïnat simultàniament.

5.3.1. Càlcul de la solució inicial

En aquesta segona manera de resolució el que es fa és resoldre un problema SALB per a cada línia de producte, de forma individual. Per a fer-ho s'utilitza una heurística greedy orientada a estacions, la qual construeix una solució en seleccionar la millor candidata factible d'una llista ordenada segons un indicador. El procediment que se segueix és el següent:

1. Càlcul del valor de l'indicador per a cada tasca i del producte j .
2. S'ordena la llista en ordre decreixent o creixent, en funció de com s'hagi definit l'indicador.
3. S'obra la primera estació, $k=1$.
4. Tenint en compte les precedències, es crea o s'actualitza la llista de candidates.

5. És selecciona la tasca i , que estigui en la posició més preferent i s'introdueix en l'estació k . No obstant això, si el temps de procés de la tasca i més el temps de les tasques incloses en l'estació pertinent supera el temps de cicle, es comprova la tasca $i+1$, i així successivament fins que cap tasca es pugui introduir.
6. Si la llista de candidates està buida, s'avança al pas 7. En cas contrari, s'obra una nova estació $k=k+1$ i es retrocedeix fins al pas 4.
7. FINAL, totes les tasques han estat assignades a una estació.

Una vegada s'han trobat totes les possibles solucions amb els 12 indicadors, es farà una nova solució que consisteix a aplicar una petita optimització a les dues línies. Per un mateix indicador i dues estacions paral·leles, és a dir, de la mateixa columna d'estacions, es mirarà si es pot compartir l'operari. Per tant, en acabar aquest pas del procediment de resolució s'obtingueran tantes solucions com indicadors es disposin. Algunes solucions seran les mateixes que les obtingudes amb els indicadors i algunes pot ser que es millori el nombre d'operaris si s'ha aconseguit compartir algun operari de la mateixa columna d'estacions.

5.3.2. Millora de la solució inicial

Entre les 12 solucions que proporciona l'heurística *greedy* amb els indicadors proposats, se selecciona la solució de millor funció objectiu. En aquest punt, s'intenta trobar millors solucions movent tasques i per a desenvolupar-ho, s'utilitzarà la metaheurística Simulated Annealing (SA). La formulació utilitzada per explicar el procediment és la que es mostra a continuació:

\bar{X}	millor solució disponible
\bar{Z}	valor de la funció objectiu de la millor solució disponible
X_{cur}	solució en curs
Z_{cur}	valor de la funció objectiu de la solució en curs
X_{vec}	solució veïna de X_{cur}
Z_{vec}	valor de la funció objectiu de la solució veïna de X_{cur}

$\prod_{(Z_{cur}, Z_{vec})}$ l'indica de probabilitat d'acceptar solucions pitjors que l'actual.

$$\prod = e^{\left(-\frac{Z_{vec} - Z_{cur}}{T}\right)}$$

T paràmetre anomenat temperatura. Cada L iteracions, el valor de T anirà disminuint geomètricament segons:

$$T = \alpha' \cdot T \text{ (si } T < 0,1 \text{ aleshores es reinicia el valor de } T \text{ al que tenia al inici).}$$

α' paràmetre reductor de la temperatura essent $0 < \alpha < 1$

L paràmetre que indica cada quantes iteracions es reduirà la temperatura

$temps_{màx}$ condició de fi, temps màxim de càlcul, en aquest procediment s'han considerat 750 segons.

Els paràmetres de la temperatura, el reductor d'aquesta i el d'iteracions s'han calibrat i s'han trobat els següents valors:

PARÀMETRES	VALOR
α'	0,85
T	130
L	260

Taula 8: Paràmetres calibrats per al Simulated Annealing. [FONT: pròpia]

El procediment a seguir s'enumera en aquests passos:

0. La solució de partida és la millor solució disponible (\bar{X}, \bar{Z}) . Per començar el procediment es defineix $X_{cur} = \bar{X}$ i $Z_{cur} = \bar{Z}$.
1. Dur a terme fins a condició de fi ($temps_{màx} = 750s$)
 - 1.1. Generació de 4 solucions veïnes factibles:
 - Intercanviar dues tasques seleccionades a l'atzar de la línia 1 (producte A).
 - Inserir una tasca escollida a l'atzar en una estació també seleccionada a l'atzar de la línia 1 (producte A).
 - Intercanviar dues tasques seleccionades a l'atzar de la línia 2 (producte B).
 - Inserir una tasca escollida a l'atzar en una estació també seleccionada a l'atzar de la línia 2 (producte B).

Cal senyalar que les tasques escollides no poden pertànyer a una mateixa estació, ni una tasca es podrà inserir en la seva pròpia estació.

- 1.2. De les 4 solucions trobades, es calculen els valors de les funcions objectius. Si una de les 4 solucions és millor que les altres 3 aleshores passarà a ser la solució veïna (X_{vec} i Z_{vec}).

D'altra banda, en cas d'empat entre dues solucions o més, es seleccionarà una de les solucions al atzar amb la mateixa probabilitat. L'opció escollida passa a ser la solució veïna (X_{vec} i Z_{vec}).

- 1.3. Si Z_{vec} no és pitjor que Z_{cur} ($Z_{vec} \leq Z_{cur}$): $X_{cur} = X_{vec}$ i $Z_{cur} = Z_{vec}$. Si a més a més Z_{vec} és millor que \bar{Z} ($Z_{vec} < \bar{Z}$), aleshores es substitueix la solució veïna per la millor solució disponible essent $\bar{X} = X_{vec}$ i $\bar{Z} = Z_{vec}$. A més a més, es tornarà a anar al pas 1.1.

En cas contrari, es genera un valor a l'atzar $p \in [0,1]$ i si p es menor que $\prod(Z_{cur}, Z_{vec})$, s'accepta la solució: $X_{cur} = X_{vec}$ i $Z_{cur} = Z_{vec}$. Si p no supera el llindar, no es modifica X_{cur} ni Z_{cur} . És en aquest pas quan reduïm el valor de la temperatura (T) si s'han realitzat totes les 260 iteracions (L) o bé la reiniciem si el valor es inferior a 0,1. Es torna al pas 1.1.

5.4. Preprocés

A continuació és mostra un preprocés que es pot fer en cas que el temps de cicle de l producte B sigui el doble que el del producte A.

Aquest preprocés consisteix a multiplicar per 2 el temps de procés de les tasques que pertanyen al producte A i establir un temps de cicle que equivaldrà al temps de cicle del producte B. Per tal de clarificar el que s'ha explicat anteriorment, podem fixar-nos en el següent exemple:

La figura 12 demostra com les tasques del producte A, assignades a una estació, es realitzaran dues vegades durant un cicle del producte B, és a dir, per dos unitats del producte A podrem fer tant sols una unitat del producte B.

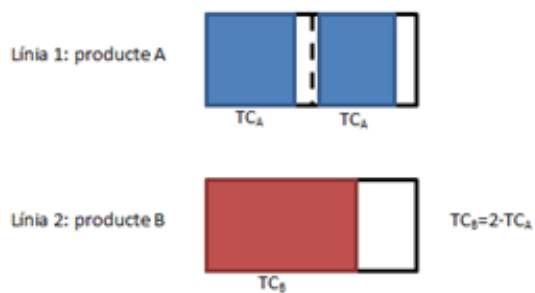


Figura 20: Exemple de la duplictat de les tasques d'A per cada cicle del producte B. [Font: pròpia]

6. Experiència computacional

6.1. Els exemplars de testeig

En l'experiment computacional, es compara els dos mètodes de resolució explicats en aquest projecte amb els diferents algorismes que han resolt aquest problema.

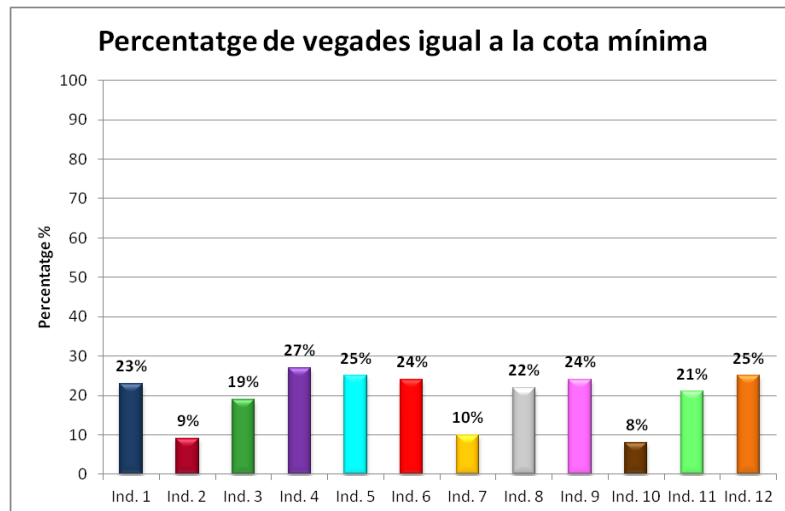
L'experiència computacional es basa en el conjunt de dades construït per Gökçen et Al. [1]. Consisteix en un total de 95 exemplars del PALBP amb dos productes per a dues línies, basat amb 14 grafs de precedències diferents que van des de 7 fins a 111 tasques. En cada cas, el graf de precedència original s'utilitza pel producte 1 mentre que pel producte 2 s'utilitza la versió original eliminant de manera aleatòria entre 1 i 4 tasques amb les relacions de precedència que comporta. Aquest conjunt de dades s'ha extret de la coneguda pàgina acadèmica sobre problemes d'equilibratge de línies i optimització: www.assembly-line-balancing.de.

Els experiments s'han realitzat a un ordinador personal amb Intel Core 2 i un processador de 2.1 Ghz amb 4GB de RAM. Pel que fa als problemes resolts amb el mètode Simulated Annealing, s'ha imposat un temps màxim d'execució del programa de 12 minuts.

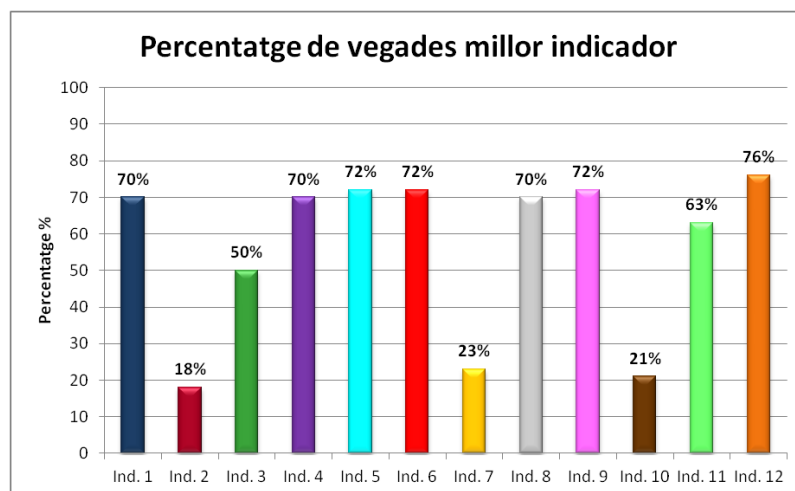
Respecte al llenguatge de programació. En els dos procediments s'ha utilitzat el visual basic (Microsoft Visual Basic 6.0)

6.2. Anàlisi dels resultats

Inicialment es mostrarà una comparativa dels diferents indicadors que s'han utilitzat per resoldre el problema. S'utilitzen dos criteris diferents per valorar els indicadors: el primer d'ells és el nombre de vegades que el resultat és igual a la fita mínima tant sols amb el valor obtingut quan es fa l'assignació de tasques amb l'indicador, i el segon és el nombre de vegades que el indicador obté la millor solució (en cas de haver-hi 2 o més indicadors amb el valor mínim, es conten tots). Aquests criteris mostren en els següents gràfics:



Gràfic 1: Percentatge de vegades que és igual a la fita mínima. [Font: pròpia]



Gràfic 2: Percentatge de vegades que és el millor indicador. [Font: pròpia]

De les dades que mostren els tres gràfics s'extreuen les següents conclusions:

- Hi ha diferents indicadors que presenten resultats molt igualats.
- Els quatre millors indicadors, de forma ajustada, són el número 5, el 6, el 9 i el 12.
- Els tres pitjors indicadors són el número 2, el 7 i el 10 de forma clara.
- Altres dades que s'han pogut extreure a l'hora de determinar els gràfics i tenint en compte tots els indicadors és que el 29% de les vegades s'aconsegueix la fita mínima calculada entre algun de tots els indicadors.

El resultat obtingut en aquest projecte (simplificat TFM) és el millor obtingut entre el procediment Ad hoc i el Simulated Annealing. El fet de que tots dos triguin molt poc temps en executar-se permet trobar dues solucions diferents i quedar-nos amb el millor valor. Aquest

valor s'ha comparat amb 3 algorismes diferents, a més a més també s'han comparat per separat els dos procediments, el Ad hoc (AH) i el Simulated Annealing (SA).

El primer d'ells és el de Gökçen (GOK) [1] el qual proposa un model matemàtic amb els exemplars que no tenen moltes tasques, ja que després el temps que es necessitaria per trobar la solució òptima és molt elevat.

El segon és el primer model que van proposar A. Scholl and N. Boysen (MOD) [6], que, de la mateixa manera que Gökçen proposen un model matemàtic que només serveix per exemplars amb poques tasques, a més a més, proposen dos objectius. Un d'ells és minimitzar el nombre d'operaris i l'altre minimitzar el nombre d'estacions intermitges buides.

Finalment, és compara amb l'algorisme de les formigues proposat per Ozbakir (mc·ACO) [4]. Aquest algorisme genera moltes solucions en paral·lel, de manera que a partir de la primera solució, cada formiga en genera de noves i així successivament.

A continuació, es mostra la taula comparativa amb els diferents algorismes. També es pot observar la solució òptima o la millor solució coneguda (MS) que s'ha trobat mitjançant mètodes exactes. En aquesta taula s'observa el valor mig d'operaris que han obtingut cada un dels diferents procediments. Les dades s'han extret dels articles dels diferents autors. A la taula no es mostra el valor obtingut en el procediment Tabu Search ja que només s'ha resolta per 55 exemplars. Al annex 1 es mostren els resultats per tots els exemplars resolts.

En aquesta taula a més a més dels valors mitjos globals, també es mostra el valor mig en diferents situacions, és a dir, en el cas que es considerin poques tasques (entre 0 i 50) en el cas de que es considerin un interval mig de tasques (entre 50 i 100) i en el cas de que considerem que hi ha gran quantitat de tasques (>100).

TASQUES	GOK	MOD	AH	SA	TFM	MS
GLOBAL	36,8	38,0	37,4	37,0	36,9	36,4
(0,50)	11,1	11,6	11,6	11,1	11,1	11,1
(50,100)	52,7	54,5	53,7	53,2	53,1	52,2
(>100)	49,3	49,6	48,6	48,3	48,3	48

Taula 9: Valors mitjos d'operaris en els diferents algorismes de resolució. [FONT: pròpia]

Com s'observa en la taula, per a exemplars petits, tots els mètodes, excepte MOD, funcionen de la mateixa manera. Si ens fixem en els exemplars mitjos, el mètode proposat en aquest treball, no va massa be, això es degut a que en els exemplars Lutz2, els resultats són dolents

comparativament. Finalment, si ens fixem en els exemplars grans, veiem que es on funciona millor, superant als demés mètodes no exactes i força prop de la millor solució trobada per mètodes exactes.

Cal afegir, que tot i que l'objectiu del problema no sigui trobar una solució en el mínim de temps possible, el mètode Ad hoc, ha necessitat una mitjana de 0,11 segons per trobar la millor solució, i el mètode Simulated Annealing 12,71 segons. Aquest segon mètode s'ha executat durant 750 segons, fet que provoca que el temps total d'execució sigui molt més elevat.

Finalment, es mostra una taula comparativa dels diferents algorismes i el nombre de vegades que s'ha trobat la fita mínima. Per trobar el valor percentual, s'ha dividit el nombre de vegades que s'assoleix la cota mínima per el total d'exemplars resolts (95 en tots els casos excepte TS que són 55)

Algorisme				GOK	MOD	TS	AH	SA	TFM	MS
Nº vegades	assoleix	fita	mínima	45	23	17	19	39	40	63
Percentatge	de	vegades	assolida	47%	24%	31%	20%	41%	42%	66%

Taula 10: Comparació dels algorismes respecte la fita mínima. [FONT: pròpia]

Com era d'esperar, els algorismes resolts de manera exacta, aporten millors solucions que les trobades en aquest projecte. Ara bé, si es compara amb el MOD, el qual només s'ha resolt de manera exacta per algorismes amb molt poques tasques, observem que la solució TFM és molt millor. Passa el mateix si ho comparem amb el TS* (resolt només per 55 exemplars), on podem afirmar que percentualment és millor el TFM.

7. Pressupost

A continuació, es detalla el pressupost per a la realització problema d'equilibrat de línies de muntatge en paral·lel considerant dues línies de muntatge i els mateixos temps de cycle o un temps el doble que l'altre.

Aquest projecte ha estat realitzat per un estudiant amb la col·laboració i supervisió d'un professor. Es tracta d'un estudi d'investigació de caràcter teòric, i que no s'ha implantat en cap línia de muntatge real. Per a l'anàlisi de costos de l'estudi s'han tingut en compte el cost dels materials i els equipaments utilitzat, així com el cost del temps dedicat.

La durada del projecte ha estat de 7 mesos, des del seu inici al març del 2016 fins a l'octubre de 2016. Durant aquest període s'han dut a terme les següents activitats: estudi de l'estat de l'art del problema, el disseny i la implementació dels algorismes, l'experimentació computacional amb el corresponent anàlisi dels resultats i finalment la realització de la memòria del projecte.

Per dur a terme el projecte s'han utilitzat diferents programes:

- Visual Basic 6.0: Programació dels diferents algorismes.
- Paquet Microsoft Office: Redacció del projecte.
- CALIBRA: Software per calibrar els diferents paràmetres.

Els materials utilitzats per a la realització del projecte han estat un portàtil HP ProBook 4510s i el diferent material d'oficina i impressions (calculadora, bolígrafs, impressora...). Per als diferents conceptes, se n'ha calculat el cost associat al projecte a partir del seu cost d'adquisició, el seu període d'amortització, i el seu temps d'utilització per al projecte.

Concepte	Cost	Cost associat al projecte
Ordinador portàtil HP ProBook 4510s	650,00 €	65,00 €
Material d'oficina i impressions	160,00 €	90,00 €
COST MATERIAL		155,00 €

Taula 11: Costs del material utilitzat. [FONT: pròpia]

A més a més, s'ha de tenir en compte el cost associat a l'enginyer. Aquest, s'ha dividit en dues parts, les hores dedicades per l'estudiant (enginyer junior) que ha desenvolupat el projecte i per altra banda les hores dedicades pel professor (enginyer sènior). En la següent taula es pot observar el cost resumit d'aquest dos conceptes:

Concepte	Temps dedicat (h)	Cost (€/h)	Cost (€)
Enginyer Junior	720 h	25,00 €/h	18.000,00 €
Enginyer Sènior	50 h	60,00 €/h	3.000,00 €
COST PERSONAL			21.000,00 €

Taula 12: : Costs dels enginyers. [FONT: pròpia]

Finalment, a la taula 11 es mostra el cost total del projecte tenint en compte els costos d'execució, el benefici suposat per al desenvolupament del projecte i l'IVA aplicable.

Cost d'execució	21.155,00€
Costos de material	155,00€
Costos de personal	21.000,00€
Benefici net enginyeria (10%)	2.115,50€
Cost sense IVA	23.270,50€
IVA(21%)	4.886,81€
COST TOTAL	28.157,31€

Taula 13: Cost global del projecte. [FONT: pròpia]

Cal destacar, que tal com s'observa en l'anterior càlcul del pressupost no es tenen en compte els costos estructurals.

Seguidament, s'estudiaran dos escenaris diferents per calcular el període de retorn del cost total del projecte. En el primer escenari es considerarà que els operaris son persones i en el segon cas es considerarà que els operaris són robots.

Primer escenari

Es considera que els operaris que es redueixen en el problema són persones. Suposant que una empresa utilitza el mètode de resolució proposat en aquest projecte i que les seves línies de muntatge passessin a tenir un treballador menys que seria el cas d'aplicar-ho en una línia amb més de 100 tasques per producte (veure taula 8), el període d'amortització seria el següent:

Si tenim en compte que el salari mig a Espanya és de 22.858,17 €/any (segons INE) i el cost total del projecte ha estat de 28.157,31 € :

$$\text{Amortització} = \frac{28.157,31 \text{ €}}{22.858,17 \text{ €/any}} = 1,23 \text{ anys} = 1 \text{ any i 3 mesos} \quad (\text{Eq. 12})$$

Segon cas

En aquest segon cas, es considera que els operaris que es redueixen en el problema són robots. Suposant que una empresa utilitza el mètode de resolució proposat en aquest projecte i que les seves línies de muntatge passessin a tenir un robot menys, el període d'amortització seria el següent:

Si tenim en compte que el cost d'un robot d'automatització pot esser de 45.000€ (ABB IRB 260) i el cost total del projecte ha estat de 28.157,31 € :

$$\text{Amortització} = \frac{28.157,31 \text{ €}}{45.000 \text{ €}} = 0,63 \text{ anys} = 7 \text{ mesos i 15 dies} \quad (\text{Eq. 13})$$

8. Impacte ambiental

Tal com s'ha especificat a la memòria del projecte, l'estudi efectuat tracta la resolució del PALBP des d'una perspectiva teòrica i d'investigació. Els procediments de resolució proposats s'han aplicat sobre exemplars genèrics de línies de muntatge i, per tant, no s'ha pogut quantificar l'impacte associat a l'aplicació sobre una línia de muntatge real.

La implicació ambiental no es veuria molt alterada si considerem que els operaris son persones. El fet de comptar amb menys operaris podria fer que el nombre d'estacions que funcionen fos inferior, i conseqüentment, el cost energètic també ho fos. Una disminució de l'energia requerida derivaria en una reducció directa de les emissions atmosfèriques i això permetria millorar la qualitat de vida de les zones industrials. Ara bé, si els operaris en comptes de esser persones fossin robots, aquest consum d'energia seria molt més elevat i per tant les emissions de CO₂ es veurien afectades de manera positiva.

Pel que fa a l'impacte econòmic, a més a més de l'estalvi energètic comentat anteriorment, hi hauria una disminució del cost pel fet de tenir menys operaris treballant en les línies de muntatge. Aquest impacte seria més elevat si en comptes de persones consideréssim que els operaris són robots.

Aquest últim fet de tenir menys treballadors, implica un augment de l'impacte social. El fet de tenir menys treballadors per fer les mateixes tasques implicaria que la càrrega de treball d'aquests fos més alta. A més a més aquests operaris de menys que treballarien a la línia de muntatge podrien ser o bé re-col·locats en alguna altra tasca o en el pitjor dels casos acomiadats.

En resum, l'impacte generat en aquest projecte només adquireix rellevància en la l'impacte econòmic, produint així un major benefici a l'empresa.

Conclusions

En el present treball es resol el PALBP mitjançant dos procediments que no s'havien utilitzat mai. El primer d'ells es basa en l'heurística Simulated Annealing i el segon es una heurística Ad hoc. L'objectiu d'aquest treball és resoldre el problema de l'assignació de tasques quan tenim dues línies en paral·lel, intentant minimitzar el nombre d'operaris utilitzats, és a dir, maximitzar l'eficiència de les dues línies de muntatge. Per a la resolució d'aquest problema s'utilitzen dos tipus de mètodes heurístics que ens permeten trobar bones solucions.

Els resultats computacionals revelen que els dos mètodes funcionen molt bé quan es comparen amb altres mètodes no exactes. Si més no, quan la comparació es fa amb resultats exactes, en algunes ocasions, el valor que s'ha trobat queda molt allunyat del millor resultat. Cal afegir també, que en el cas de tenir un nombre molt gran de tasques, els dos mètodes proposats funcionarien amb la mateixa rapidesa, ja que quasi no hi ha hagut alteració d'interval de temps des del primer exemplar (7 tasques) fins a l'últim exemplar (111 tasques). En el cas del mètode Ad hoc, a resolt tots els exemplars amb un temps mig de 0,11 segons i el mètode Simulated Annealing amb un temps mig de 12,71 segons.

A més a més, també s'observa que en els últims exemplars que es quan major nombre de tasques hi ha, els dos procediments funcionen millor. No sols funcionen millor sinó que la distància respecte la mitjà de la millor solució trobada és molt petita. Malauradament, no s'ha pogut comparar amb exemplars amb major nombre de tasques ja que no han estat resolts pels demes autors.

Pel que fa a cada un dels dos mètodes proposats, s'han utilitzats 12 indicadors per resoldre els exemplars. Aquest fet ens permet observar que l'indicador que millor funciona en l'equilibratge de línies és el que combina el temps de procés de la tasca, més tots els temps de procés de les seves tasques successores.

Per concloure, s'ha observat que les empreses podrien estalviar-se diners si apliquessin aquest mètode d'equilibratge en les seves línies. El cost d'execució del projecte s'amortitzaria ràpidament tant en el cas de que els operaris fossin persones (1 any i 3 mesos) com en el cas de que fossin robots (7 mesos i mig).

Agraïments

En acabar el llarg trajecte que és l'elaboració d'un TFM i mirar enrere, ens adonem que són moltes i diverses les aportacions rebudes. Seria injust no reconèixer que aquest treball també és fruit de la col·laboració de moltes persones a les quals cal agrair el seu suport i ajut, i de manera especial al tutor d'aquest treball, el professor Alberto García, així com a les persones més properes al meu entorn, com la meva família i la meva parella, els quals sense el seu suport no hagués estat possible tirar endavant aquest projecte.

A tots i a totes de tot cor: moltes gràcies!

Bibliografia

Referències bibliogràfiques

- [1] Gökçen, H., Ağpak, K. and Benzer, R. (2006). Balancing of parallel assembly lines. *International Journal of Production Economics*, 103(2), pp.600-609.
- [2] U. Ozcan, H. Cercioglu, H. Gokcen and B. Toklu (2009). A tabu search algorithm for the parallel assembly line balancing problem, *Gazi University Journal of Science* 22 (4) 313–323.16
- [3] R. Benzer, H. Gokcen, T. Cetinyokus and H. Cercioglu (2007). A network model for parallel line balancing Problem, *Mathematical Problems in Engineering: Article ID 10106*, Vol. 2007, doi:10.1155/2007/ 10106.
- [4] Ozbakir, L., Baykasoglu, A., Gorkemli, B. and Gorkemli, L. (2011). Multiple-colony ant algorithm for parallel assembly line balancing problem. *Applied Soft Computing*, 11(3), pp.3186-3198.
- [5] A. Scholl and N. Boysen (2008). The multiproduct parallel assembly lines balancing problem: model and optimization procedure, *Working and Discussion Paper Series School of Economics and Business Administration Friedrich-Schiller-University Jena*, ISSN 1864-3108.
- [6] Scholl, A. and Boysen, N. (2009). Designing parallel assembly lines with split workplaces: Model and optimization procedure. *International Journal of Production Economics*, 119(1), pp.90-100.
- [7] Lale Ozbakira, Adil Baykasoglu, Beyza Gorkemlia, Latife Gorkemlia (2011) “Multiple-colony ant algorithm for parallel assembly line balancing problem”. *Applied Soft Computing* Vol.11, pp. 3186–3198.
- [8] Martino, L. and Pastor, R. (2010). Heuristic procedures for solving the general assembly line balancing problem with setups. *International Journal of Production Research*, 48(6), pp.1787-1804.

- [9] Corominas, A. and Pastor, R. (2010). Designing greedy algorithms for the flow-shop problem by means of Empirically Adjusted Greedy Heuristics (EAGH). J Oper Res Soc, 62(9), pp.1704-1710.
- [10] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics, 21(6), p.1087.
- [11] Kirkpatrick, S., C. D. Gelatt Jr., M. P. Vecchi (1983). Optimization by Simulated Annealing, Science, 220, 4598, p671-680.
- [12] C.G.E. Boender, A.H.G. Rinnooy Kan, L. Stougie, G.T. Timmer (1982). A Stochastic Method for Global Optimization. Mathematical Programming vol. 22 pp. 125-14.

ANNEX 1: Resultats

A continuació es mostra la taula de resultats dels diferents exemplars que s'han comparat en aquest projecte. A més a més del nombre d'operaris de cada un dels procediments, també es mostra la fita mínima de cada exercici, el temps de cicle (TC), el nombre d'estacions utilitzades (K*) i el nombre de tasques que hi ha en cada una de les línies.

Problema	Nº tasques línia 1/ línia 2	TC	K*	Fita mínima	GOK	MOD	TS	AH	SA	TFM	MS
Mertens	7/6	9	4	6	7	7	-	7	7	7	7
		11	3	5	5	5	-	5	5	5	5
		13	3	5	5	5	-	5	5	5	5
		17	2	4	4	4	-	4	4	4	4
Jaeschke	9/8	9	5	8	8	8	-	8	8	8	8
		11	4	7	7	7	-	8	7	7	7
		13	3	6	6	6	-	6	6	6	6
		15	3	5	5	5	-	5	5	5	5
		17	3	4	4	4	-	5	4	4	4
Jackson	11/10	8	7	11	13	13	-	13	13	13	13
		10	5	9	9	9	-	11	9	9	9
		13	4	7	7	7	-	8	7	7	7
		15	4	6	6	6	-	7	6	6	6
		19	3	5	5	5	-	5	5	5	5
Roszieg	25/24	14	10	18	18	20	-	19	18	18	18
		16	8	16	16	16	-	17	16	16	16
		17	8	15	15	16	-	16	15	15	15
		22	6	12	12	12	-	12	12	12	12
		30	5	9	9	9	-	9	9	9	9
Sawyer	30/28	25	14	26	26	28	-	27	26	26	26
		27	13	24	25	26	-	25	24	24	24
		30	12	22	22	23	-	23	22	22	22
		36	10	18	18	19	-	19	18	18	18
		41	8	16	16	16	-	16	16	16	16

		54	7	12	12	12	-	13	12	12	12
Kilbridge	45/43	57	10	20	20	20	20	20	20	20	20
		79	7	14	14	15	14	14	14	14	14
		92	6	12	12	12	12	12	12	12	12
		110	6	10	10	10	11	10	10	10	10
		138	4	8	8	8	8	8	8	8	8
		184	3	6	6	6	6	6	6	6	6
Hahn	53/51	2004	8	14	14	17	14	15	14	14	14
		2338	7	12	12	15	12	12	13	12	12
		2806	6	10	10	11	10	11	10	10	10
		3507	5	8	8	9	8	9	8	8	8
		4676	4	6	6	7	6	7	6	6	6
Tonge	70/66	160	23	43	45	46	-	45	45	45	44
		168	22	41	43	45	-	44	44	44	42
		207	18	34	34	35	-	35	35	35	34
		234	16	30	30	32	-	31	31	31	30
		270	14	26	26	27	-	27	27	27	26
		293	13	24	24	25	-	25	25	25	24
Wee-mag	75/71	28	63	105	123	123	123	123	123	123	123
		29	63	102	123	123	123	123	123	123	123
		31	62	95	121	122	121	121	121	121	121
		33	61	89	119	120	119	119	119	119	119
		34	61	87	119	120	119	119	119	119	119
		41	59	72	116	116	116	116	116	116	116
		42	55	70	107	108	107	108	107	107	107
		43	50	69	98	98	98	98	98	98	98
		49	32	60	62	63	63	62	62	62	62
		54	31	55	60	61	60	61	60	60	60
Arcus1	83/79	3786	21	39	40	42	40	41	40	40	40
		3985	20	37	38	40	38	39	38	38	37
		4206	19	35	36	38	36	36	36	36	36
		4454	18	33	34	35	34	35	34	34	34
		4732	17	31	32	33	32	33	32	32	32
		5853	14	25	26	26	26	26	26	26	25
		6842	12	22	22	23	22	22	22	22	22
		7571	11	20	20	21	20	20	20	20	20
		8412	10	18	18	19	18	18	18	18	18
		10816	8	14	14	14	14	15	14	14	14

Lutz2	89/85	11	49	86	90	97	-	97	95	95	88
		12	44	79	82	88	-	87	85	85	79
		13	40	73	75	81	-	79	77	77	73
		14	37	68	71	79	-	70	72	70	68
		15	34	63	64	68	-	64	65	64	63
		16	31	59	60	66	-	64	61	61	59
		17	29	56	56	60	-	60	58	58	56
		19	26	50	50	53	-	52	51	51	50
		20	25	47	48	51	-	50	50	50	47
Lutz3	89/85	75	23	43	45	46	44	46	46	46	44
		79	22	41	43	45	42	43	42	42	41
		83	21	39	40	43	40	41	40	40	39
		87	20	37	38	40	38	40	40	40	37
		92	19	35	36	37	36	37	36	36	35
Mukherje	94/90	176	25	47	48	49	48	48	48	48	47
		183	24	45	46	47	46	46	46	46	45
		192	23	43	44	44	44	44	44	44	43
		201	22	41	42	44	42	43	42	42	41
		211	21	39	40	40	40	40	40	40	39
		222	20	37	38	40	38	38	38	38	37
		234	19	35	36	37	36	36	36	36	35
		248	18	33	34	35	34	35	34	34	33
		263	17	31	32	33	32	33	32	32	32
		281	16	29	30	31	30	31	30	30	29
		301	15	27	28	29	28	29	28	28	28
		324	14	26	26	27	26	27	26	26	26
		351	13	24	24	25	24	25	24	24	24
Arcus2	111/107	5785	28	52	55	54	54	54	54	54	53
		6016	27	50	53	52	51	52	51	51	51
		6267	26	48	50	50	50	50	50	50	49
		6540	25	46	48	50	48	47	47	47	47
		6837	24	44	46	48	45	46	45	45	45
		7162	22	42	44	44	43	43	43	43	43

Taula 14: Comparació dels diferents resultats obtinguts amb tots els algorismes que s'han resolt a la literatura.
[FONT: pròpia]

ANNEX 2: Documentació adjunta

A més a més de la memòria i els annexes, també es presenta la següent informació.

- Enunciats dels exemplars resolts.
- Codi dels dos procediments proposats.
- Resultats dels exemplars resolts mitjançant els dos procediments.